



Universidad de las Ciencias Informáticas

Dirección de Informatización

Diseño de un catálogo de servicios Web basado en las especificaciones y normas de la UDDI para la plataforma de informatización en la UCI.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

AUTORES

Elisa Clavero Vázquez

Danielis Tur Sivila

TUTOR

Ing. Yunier Saborit Ramírez

Ciudad de La Habana, Cuba

Junio, 2007

DECLARACIÓN DE AUTORÍA

Declaramos, Elisa Clavero Vázquez y Danelis Tur Sivila ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los __ días del mes de junio del 2007.

Elisa Clavero Vázquez

Danelis Tur Sivila

Ing. Yunier Saborit Ramírez

Agradecimientos

A nuestros padres y familiares que nunca nos han desamparado y siempre serán la luz primera a lo largo del camino de nuestras vidas. A nuestros amigos que nunca podremos olvidar porque ya forman parte de nosotras. Un agradecimiento especial a nuestro tutor por brindarnos tanto apoyo y por tener tanta paciencia en el desarrollo del trabajo. A todos los que de una forma u otra han contribuido con nuestra formación en estos cinco años y a los que colaboraron en el desarrollo de este trabajo.

Gracias a todos.

Dedicatoria:

A quienes me forjaron y me han guiado a lo largo de mi vida con mucho amor y por darme fuerzas en todo momento, Mi madre y mis abuelos.

*A quien siempre me ha dado su apoyo, cariño y es como un hermano para mí
Mi primo Osmani.*

A quienes me acompañan siempre y hacen que mi vida se torne mejor con sus consejos y acciones.

Mis Tíos Melso, Pimentel y Antonio, Mis primo Jorge, Yoamí y Oslem, Mis Tías Arelis, Gladis y Marlene.

A mi novio Antonio por ser mi amigo y compañero y por hacer que mi vida tenga un mejor sentido.

A mis amigos que a pesar de estar lejos de mi familia, su compañía hace que no me sienta lejos de casa.

Elisa Clavero Vázquez.

A quienes me criaron, me educaron y me dieron todo el amor del mundo, Mi abuela y Mis Padres.

A mi hermano Einer por su apoyo y preocupación.

A toda mi familia en general en especial a mi Tía Norma y mis Primas Mayumi y Yinkra.

A quienes en estos cinco años han convivido conmigo dándome apoyo y cariño convirtiéndose en parte de mi vida,

A todos mis amigos en especial Maiquel, Tomás, Liudmila, Yanet, Yani, Nina, Leo y especialmente a Elisa.

A mi novio Omar por el apoyo y el amor brindado durante este tiempo.

Dianelis Tur Sivila.

RESUMEN

El presente trabajo se desarrolló en la Universidad de las Ciencias Informáticas, centro en el cual se desarrollan una gran cantidad de aplicaciones Web, tanto para el desarrollo de la universidad como para la exportación.

Este trabajo se realiza con el objetivo de lograr un software que brinde interoperabilidad, organización y publicación de los servicios Web que se implementan en las diferentes aplicaciones que se desarrollan en el centro. Para ello se estructura en 4 capítulos donde se recoge la fundamentación teórica del trabajo, el modelo del sistema, la descripción de la solución propuesta y el estudio de factibilidad.

Se propone como solución el diseño de un sistema capaz de almacenar la información referente a los servicios web así como darlos a conocer; siguiendo los estándares internacionales del catálogo de servicios web denominado UDDI (Descripción Universal, Descubrimiento e Integración). .

ÍNDICE

INTRODUCCIÓN.....	1
Fundamentación Teórica	4
1.1 Introducción.....	4
1.2 Estado de UDDI a nivel internacional.....	4
1.3 Estado de UDDI a nivel nacional.....	7
1.4 Tendencia.....	8
1.4.1 SOA.....	8
1.4.2 Web Service	10
1.4.4 SOAP.....	13
1.4.5 WSDL.....	15
1.4.6 Seguridad.....	16
1.5 La arquitectura de informatización de la UCI y la implementación de UDDI.	18
1.5.1 Las aplicaciones Web.	19
1.5.2 Modelo Cliente Servidor.....	20
1.5.3 PHP (PHP: Hypertext Preprocessor).....	21
1.5.4 Servidor Web Apache.	22
1.5.5 AJAX.....	22
1.5.6 Patrones de Diseño.....	24
1.5.7 Modelo Vista Controlador (MVC).....	24
1.5.8 Sistemas de Gestión de Base de Datos.....	26
1.5.8.1 PostgreSQL.....	26
1.5.9 Proceso de Desarrollo.....	28
1.5.10 Herramientas utilizadas.....	29
1.6 Conclusiones.....	31
Modelo del Sistema	33
2.1 Introducción.....	33
2.2 Descripción de los proceso del negocio propuestos.....	33
2.3 Modelo del Dominio.....	35
2.3.1 Conceptos:	35
2.3.2 Diagrama del modelo de dominio.....	36
2.4 Requerimientos	37
2.4.1 Requerimiento funcional.....	37
2.4.2 Requerimientos no funcionales	39
2.5 Identificación de actores.....	42
2.6 Paquetes y sus relaciones.....	43
2.6.1 Paquetes Autenticación.....	43

2.6.2	Paquetes Búsqueda y solicitud de información.....	45
2.6.3	Paquetes Gestión de información.....	54
2.6.4	Paquetes Gestionar Administración.....	65
2.7	Conclusiones.....	82
Análisis y Diseño		83
3.1	Introducción.....	83
3.2	Modelo de Análisis	83
3.3	Modelo de Diseño.....	87
3.3.1	Diagramas de Clases de Diseño.....	88
3.3.2	Diagramas de interacción.....	92
3.4	Principios de Diseño.....	92
3.4.1	Interfaz de usuario.....	92
3.4.2	Tratamiento de errores.....	93
3.5	Diseño de la base de datos.....	95
3.5.1	Diagrama de clases persistentes.....	95
3.5.2	Modelo de datos.....	97
3.6	Diagrama de Despliegue.....	97
3.7	Conclusiones.....	98
Estudio de Factibilidad.....		100
4.1	Introducción.....	100
4.2	Planificación basada en casos de uso.....	100
4.3	Beneficios tangibles e intangibles.....	107
4.4	Análisis de costos y beneficios.....	107
4.5	Conclusiones.....	108
CONCLUSIONES.....		109
RECOMENDACIONES.....		111
REFERENCIA BIBLIOGRÁFICA		112
BIBLIOGRAFÍA.....		114
ANEXOS.....		117
Anexo 1: Modelo Cliente Servidor de tres Capas.....		117
Anexo 2 Ejemplo mensaje SOAP petición.....		118
Anexo 3 Ejemplo mensaje SOAP respuesta.....		118
Anexo 4 Ventajas de WSDL.....		119
Anexo 5 Interoperabilidad Servicios Web en PHP		119
Anexo 6 Diagrama de Clases del Modelo de Diseño.....		120
TABLAS.....		124
Tabla 1 Descripción de las tablas de la Base de datos.....		124
GLOSARIO DE TÉRMINOS Y SIGLAS.....		141

INTRODUCCIÓN

Con el desarrollo de arquitecturas como SOA, que ha conducido a una nueva y progresiva era de la información, en el mundo actual se ha introducido una problemática con las aplicaciones Web debido al surgimiento de problemas entre los que se perfilan: desconocimiento de la existencia de un Servicio Web dado, lo que provoca la re implementación innecesaria de este. Pérdida de Servicios WEB. La no existencia de un sitio en el que estén ubicados y ordenados los servicios para la publicación y búsqueda de estos. Mala categorización de los servicios Web.

Se propone una arquitectura que permita desarrollar una UDDI que se adapte a las necesidades y requerimientos de las empresas debido a las ventajas que proporciona para el desarrollo de aplicaciones Web. El problema a resolver se basa en la no existencia de una UDDI que se adapte a las características de la arquitectura que se propone para el desarrollo de software.

El creciente desarrollo de aplicaciones Web en las empresas, tanto para el desarrollo dentro de aplicaciones internas como para software de exportación, y la necesidad de interacción entre ellos para lograr un mayor desarrollo y rápido intercambio a través de los servicios Web implementados en los mismos, demanda un software que sea capaz de lograr la interoperabilidad, organización y publicación de los servicios Web implementados en las diferentes aplicaciones. El **problema** se puede formular entonces de la siguiente manera: ¿Cómo resolver el problema del manejo, organización y publicación de los servicios Web desarrollados sobre la arquitectura basada en SOA?

Como antecedentes de este trabajo se tienen todas las especificaciones que OASIS determina el estándar para la realización de una UDDI, incluyendo la versión 3.0.2 UDDI Spec Technical Committee, documento que describe cuales son sus características y funciones a tener en cuenta para implementar una UDDI. Otro antecedente es Estudio de UDDI, documento de diploma realizado en la UCI.

El **objeto de estudio** de este trabajo son las aplicaciones UDDI para la gestión de la interoperabilidad, organización y publicación de servicios Web en el contexto de la arquitectura basada en SOA en las empresas. Mientras **el campo de acción** es la UDDI como ente gestor de la interoperabilidad en la propuesta de Arquitectura basada en SOA para los entornos empresariales.

Este trabajo tiene como objetivo general el Diseño del Catálogo de Servicio Web en software libre que cumpla con las especificaciones y estándares internacionales UDDI.

Como **objetivos específicos** se plantean los siguientes:

- Desarrollar el análisis y diseño del sistema acorde con las especificaciones UDDI.
- Establecer definiciones que cumplan con estándares internacionales.
- Obtener el diseño de una base de datos capaz de almacenar de manera organizada la información que se manipula.
- Obtener un prototipo de interfaz de usuario.

Para cumplir los objetivos se desarrollaron las siguientes **tareas**:

- Investigar y resumir las especificaciones de UDDI.
- Realizar el análisis y diseño del sistema.
- Documentar la información referente al análisis y diseño del sistema, así como todo lo referente al flujo de trabajo.
- Diseñar una base de datos que soporte la mayoría de las funcionalidades del sistema.
- Agrupar los estándares de seguridad establecidos para los servicios web.
- Definir los protocolos que se utilizan para transportar la información.

La presente propuesta contribuye a mejorar y a optimizar las aplicaciones desarrolladas en nuestra universidad logrando diseñar una herramienta para registros distribuidos de información sobre servicios web. Así como lograr que sea usado tanto por programas como por programadores para encontrar información acerca de servicios y para saber cómo invocarlos y trabajar con ellos.

De esta forma se garantiza el descubrimiento y acceso a servicios que brindan las aplicaciones a través de los servicios web sin necesidad de interactuar directamente con estas y poder usar la información.

El presente trabajo, estructurado en 4 capítulos, resume la siguiente información:

Capítulo 1. Fundamentación Teórica: descripción del objeto de estudio, sistemas existentes vinculados al campo de acción, tendencias y tecnologías actuales seleccionadas a emplear en el desarrollo de la propuesta y por qué su utilización.

Capítulo 2. Modelo del sistema: describe el negocio a través de un modelo de Dominio, y se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente.

Capítulo 3: Análisis y diseño: plantea los detalles relacionados con el análisis y diseño del sistema que se propone, se utilizan para su modelado los diagramas de clases del análisis y diseño web, de interacción para el diseño web, y el diseño de la base de datos que se necesita para el almacenamiento de la información persistente.

Capítulo 4: Estudio de factibilidad: estudio de factibilidad económica realizado para la universidad, en el que se determina si es factible o no el desarrollo del software propuesto, analizando los diferentes criterios que influyen en el cálculo del esfuerzo, tiempo de desarrollo y costo para la universidad.

CAPÍTULO **1**

Fundamentación Teórica

1.1 Introducción.

Este capítulo contiene los principales problemas que fundamentan la propuesta de solución, y los objetivos generales y específicos que se persiguen. Además de brindar un enfoque general de acerca de UDDI. Se describen además las tecnologías actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta.

1.2 Estado de UDDI a nivel internacional.

1.2.1 UDDI

El tema UDDI y como tal su estado a nivel internacional es muy joven. Las primeras palabras e ideas acerca de UDDI emergen a finales de la década de los 90, como respuesta a una serie de preguntas formuladas sobre como manejar y administrar de manera eficiente y dinámica la creciente y gran cantidad de software que se basan en los Servicios Web existentes, algunas de estas preguntas eran las siguientes:[1]

- *¿Cómo se clasifica la información de forma coherente?*
- *¿Cómo se descubren los servicios Web?*
- *¿Cómo se puede interactuar en tiempo de ejecución con este mecanismo de descubrimiento cuando mi aplicación depende de un Servicio Web?*
- *¿Cómo repercute esto en la localización?*

Luego UDDI surge del esfuerzo y la cooperación que durante varios meses representantes de Microsoft, IBM y otras compañías brindaron entre si, iniciando en el verano del 2000 y en los primeros días de septiembre de ese mismo año se divulga oficialmente su creación. A partir de este punto, se puede apreciar que UDDI cuenta hoy en día con un soporte bastante amplio en cuanto a compañías que se han unido al empresa, hasta el momento relaciona a una comunidad de 26

compañías que juntaron sus esfuerzos para desarrollar una especificación basada en estándares abiertos y tecnologías no propietarias que permitiera resolver los retos anteriores. [1]

El objetivo de UDDI es facilitar el descubrimiento de servicios tanto dinámicamente como en tiempo de diseño. La primera implementación pública fue dada a conocer el 2 de mayo del 2001, este registro (UDDI) es conocido como el Registro de Negocios UDDI (UDDI Business Registry), actualmente esta compuesto por cuatro registro replicados, que son mantenidos por Microsoft, IBM, SAP y NTT Communications y son llamados operadores UDDI, es un registro empresarial global en el que los usuarios pueden realizar búsquedas y publicaciones sin coste alguno. La gran cantidad de compañías que ya forman la base esta tecnología no está limitada y se espera que otras compañías decidan unirse al grupo de operadores, con el objetivo de que el usuario pueda seleccionar con criterios, como calidad de servicio o facilidad de uso de la interfaz Web, que todo operador esté obligado a implementar. [1]

A partir de la creación de esta infraestructura para servicios Web, los datos sobre estos servicios se pueden encontrar de forma sistemática y confiable en un instrumento universal totalmente independiente de proveedores. Se pueden llevar a cabo búsquedas categóricas precisas utilizando sistemas de identificación y taxonómicos extensibles. La integración de UDDI en tiempo de ejecución se puede incorporar a las aplicaciones. Como resultado, se fomenta el desarrollo de un entorno de software de Servicios Web. [1]

1.2.2 ebXML

ebXML (Electronic Business XML Working Group) ha sido coordinado por la UN/CEFACT y OASIS desde 1998. Es un conjunto de especificaciones que describen los documentos (en XML) y los procesos más habituales entre organizaciones que desean utilizar el comercio electrónico, a través de Internet, de una forma confiable. Se encuentra dirigido principalmente a cualquier empresa de cualquier tamaño sin importar su ubicación.

La especificación de mensajería de ebXML está basada en SOAP con Attachments y no usa WSDL, agregando varias calidades de servicio, como la seguridad, mensajería garantizada y complacencia con los modelos de interacción de procesos de negocio (B2B).

La arquitectura de ebXML comienza con un proceso comercial y un modelo de información, una estructura de los esquemas de XML, y la definición de los requisitos para las aplicaciones que procesan los documentos y los intercambian entre los socios comerciales.

- La arquitectura de ebXML se centra en las interacciones orientadas a documentos.
- Tecnologías adicionales están enfocadas primordialmente en las siguientes áreas:
 - Seguridad
 - Flujo de procesos
 - Transacciones
 - Messaging
- El aspecto más importante es el relativo a seguridad, el cual, en algunas tecnologías adicionales a Web services, incorporan tecnologías relacionada a este aspecto.

1.2.3 Diferencia entre UDDI y ebXML

La diferencia fundamental entre UDDI y el ebXML es que UDDI está apuntando crear un registro estándar para las compañías que aceleran la integración de sistemas alrededor de los mercados netos, mientras que el ebXML está trabajando para estandarizar cómo XML es usada en la integración general del negocio-a-negocio (B2B).

La base del modelo de UDDI por lo tanto se centra particularmente en conectividad del middleware, y usar XML para describir los sistemas que las compañías utilizan interconectar el uno con el otro. Planes de UDDI para hacer esto es almacenar la información sobre los perfiles y las capacidades de sociedades de la integración en un directorio compartido en el cual otras compañías pueden tener acceso a un sistema de estándares de XML que son trabajados actualmente.

1.2.4 WSIL

WSIL (*Web Service Inspection Language*) es una especificación muy reciente iniciada por IBM y Microsoft que ofrece un mecanismo complementario a UDDI y ebXML para encontrar servicios web en una red como Internet. Existen dos razones principales por las que surgió WSIL:

- Falta de Moderación : Debido a la misma escalabilidad con la que fue diseñado UDDI, existe una falta para moderar los servicios Web que son publicados en este tipo de Directorios, lo

anterior trae consigo la publicación de "Web Services Description Language/(WSDL)" inválidos, la duplicidad de WSDL e inclusive la publicación de servicios Web no disponibles.

- Falta de Calidad de Servicio (QoS): Este punto aunque relacionado con el anterior se refiere a la garantía del servicio ofrecido por el proveedor del servicios web esto es, debido a que UDDI es un directorio público en Internet, quién garantiza que el servicios web utilizado o comprado cumpla con determinados requerimientos. Esto nos lleva al antiguo concepto de negocios: Comprar o adquirir a quien ya conocemos, y es precisamente en este principio en el que esta basado WSIL.

A diferencia de UDDI donde se realizan búsquedas en un directorio centralizado, mediante WSIL se inspecciona un "Web-Server" conocido realizando una búsqueda por web services y es mediante archivos escritos en WSIL que se logra su descubrimiento.

1.3 Estado de UDDI a nivel nacional.

El nivel del estado de las UDDI en Cuba, es prácticamente cero. Existen varios factores por la que la llegada de esta tecnología es un poco tardía, pero cabe destacar que a partir de los primeros pasos de los desarrolladores cubanos en el campo de la tecnología de los servicios Web, se comienza a valorar y pensar en como tener un buen control de estos servicios, de manera que se decide tener en cuenta la tecnología UDDI, como potente herramienta para la administración de estos servicios.

Dos puntos significativos a señalar en cuanto al estado de esta tecnología, es que Cuba no cuenta con ningún grupo de trabajo u organización científica que pertenezca a la Comunidad UDDI, lo que facilitaría indiscutiblemente el desarrollo de este campo y el otro punto no menos importante, es la no realización de una investigación profunda sobre esta nueva tecnología.

La presente propuesta se basa en el análisis y diseño de un sistema que sea capaz de almacenar la información acerca de los servicios web, así como darlos a descubrir. Basándose en la especificaciones internacionales de UDDI como protocolo estándar en la arquitectura orientada a servicios. Garantizando que las empresas cubanas puedan agrupar sus servicios web en un registro, y dando un paso más hacia la interoperabilidad entre los sistemas, tema importante hoy en día para el desarrollo de las aplicaciones.

A continuación se explican elementos importantes para el desarrollo del catálogo de servicios web.

1.4 Tendencia.

Cuando emana una nueva tendencia, es difícil explicarla, y en particular, tratar de que quienes reciban la explicación, perciban la importancia de la tendencia y las implicaciones que tiene para sus intereses.

La tecnología UDDI es una nueva y poderosa herramienta que contribuye al desarrollo de los sistemas distribuidos y está encaminada a proporcionar una plataforma universal para la integración de las transacciones realizadas mediante Servicios Web.

La comunicación entre los negocios a nivel de aplicaciones siempre ha sido un reto, dada la vasta existencia de plataformas, herramientas, mecanismos, y procesos que cada quien utiliza. La popularidad reciente del **XML** (**eXtensible Markup Language**) en estos sitios, promete una solución para el intercambio de datos de una forma transparente. El XML permite construir todo tipo de definiciones de sintaxis de datos y permite que la información sea manipulada por las aplicaciones, eliminando todo tipo de adivinanza en las comunicaciones. Se pueden acordar las etiquetas a utilizar, definir las interfaces, y el procesamiento de datos puede ser estandarizado. Los servicios Web son programas componentes reutilizables que usan XML como marco de referencia de comunicación extensible estándar para facilitar este tipo de comunicación de máquina-a-máquina. [1]

1.4.1 SOA

La Arquitectura Orientada a Servicios, es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario.

En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado. La mayoría de las definiciones de SOA identifican la utilización de servicios Web empleando SOAP y WSDL en su implementación, no obstante se puede implementar una SOA utilizando cualquier tecnología basada en servicios.

Al contrario de las arquitecturas orientado a objetos, las SOAs están formadas por servicios de aplicación débilmente acoplados y altamente inter operables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma subyacente y del lenguaje de programación la cual es WSDL. [2]

1.4.1.1 Diseño y desarrollo de SOA

La metodología de modelado y diseño para aplicaciones SOA se conoce como análisis y diseño orientado a servicios. La arquitectura orientada a servicios es tanto un marco de trabajo para el desarrollo de software como un marco de trabajo de implantación. Para que un proyecto SOA tenga éxito los desarrolladores de software deben orientarse ellos mismos a esta mentalidad de crear servicios comunes que son orquestados por clientes o middleware para implementar los procesos de negocio. El desarrollo de sistemas usando SOA requiere un compromiso con este modelo en términos de planificación, herramientas e infraestructura. [2]

Cuando se habla de una arquitectura orientada a servicios se refiere a un juego de servicios residentes en Internet o en una intranet, usando servicios web. Hay un juego de estándares de los que se habla ligados a los servicios web. Incluyen los siguientes: [2]

- XML
- HTTP
- SOAP
- WSDL
- UDDI

1.4.1.2 Características de la arquitectura SOA para UDDI

- **Visibilidad:** Un negocio no puede utilizar lo que no puede encontrar. No puede manejar lo que no puede identificar. Las organizaciones crecen cada vez más y pueden llegar a implementar múltiples sistemas que exponen información redundante, creando un conflicto entre las organizaciones. SOA está sobre ganar visibilidad en los funcionamientos de los varios usos que una compañía construye y despliega. La visibilidad evita usos redundantes, y permite la comunicación a través de una organización. [3]
- **Flexibilidad:** Los servicios son componentes de *software* con interfaces bien definidas e independientes de su implementación. Un aspecto importante de SOA es la separación entre la interfaz de servicio (el qué) y su implementación (el cómo). A los clientes que consumen dichos servicios no les interesa conocer la forma en que estos responden a sus solicitudes. [3]
- **Reusabilidad:** En realidad, el despliegue de SOA en una organización es un ejercicio de creación de servicios a partir de lo existente en los sistemas IT y la infraestructura. Los

servicios deben ser auto contenido, posibilitando que sean invocados directamente por los clientes cuando sea necesario. [3]

- Adaptabilidad. Una ventaja de la reutilización es la puesta en práctica de los servicios de negocio débilmente acoplados que se pueden reutilizar, componer y recomponer para resolver nueva necesidad del negocio. Logrando que se alcancen tiempos de desarrollo rápidos para la aplicación; una consistencia y una confiabilidad más rápida para utilizar servicios web. [3]
- Interoperabilidad: La interoperabilidad es tal vez el principio más importante de SOA. Como método de implementación de SOA, los servicios web deben ofrecer importantes beneficios de interoperabilidad, y permitir la ejecución de servicios web distribuidos en múltiples plataformas de software y arquitecturas de hardware. Permitiendo que las aplicaciones se puedan comunicar a través de los servicios web que implementan, independientemente del sistema operativo y el lenguaje sobre las que corran. [3]

1.4.2 Web Service

El término Web Services describe una forma estandarizada de integrar aplicaciones Web mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de Internet, XML es usado para describir los datos, SOAP se ocupa para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se ocupa de dar a conocer cuales son los servicios disponibles. Uno de los usos principales es permitir la comunicación entre empresas, y entre las empresas y sus clientes. Los Web Services permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información. [4]

Los Servicios Web permiten que diferentes aplicaciones, de diferentes fuentes, se comuniquen sin tener que hacer desarrollos complejos, y como se comunican en XML, no están restringidos a un sistema operativo ni a una tecnología específica.

El concepto de los Web Service es el comienzo de una nueva arquitectura para construir sistemas orientados a los servicios. El cambio de un sistema orientado a objetos a uno orientado a los servicios es una idea que evolucionó de la nueva perspectiva planteada por la red global de Internet. Los Servicios Web es un nuevo paradigma en el desarrollo de sistemas distribuidos que proveerá una plataforma para todas las transacciones de negocio a negocio (B2B) en la Internet. [4]

1.4.2.1 Ventajas de los servicios Web. (Nick A. Monghert)

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.4.2.2 Middleware

El Middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores. [5]

Estos han aparecido de manera relativamente reciente en el mundo de la informática, facilitando la computación distribuida, mediante conexión de múltiples aplicaciones para crear una mayor, sobre una red. [5]

El Middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware. [5]

Por lo general el middleware del lado cliente está implementado por el Sistema Operativo subyacente, el cual posee las librerías que implementan todas las funcionalidades para la comunicación a través de la red. Una de las clasificaciones del middleware el protocolo RPC. [5]

1.4.3 RPC

El RPC, Llamada a Procedimiento Remoto, es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre

ambos. El protocolo es un gran avance sobre los sockets usados hasta el momento. De esta manera el programador no tenía que estar pendiente de las comunicaciones, estando éstas encapsuladas dentro de las RPC. [6]

Las RPC son muy utilizadas dentro del paradigma cliente-servidor. Siendo el cliente el que inicia el proceso solicitando al servidor que ejecute cierto procedimiento o función y enviando éste de vuelta el resultado de dicha operación al cliente. [6]

1.4.3.1 XMLRPC y SOAP

Hoy en día hay dos grandes tendencias: XML-RPC y SOAP para el uso de servicio web. A la hora de programar un servicio web, hay que decidir qué protocolo usar, ya que un protocolo es incompatible con el otro. De modo que si programamos nuestro servicio web con XML-RPC, no podremos invocarlo desde un lenguaje de programación que trabaje con SOAP, como por ejemplo .Net de Microsoft.

A través de XMLRPC/SOAP se logra que el intercambio de información sea llevado a cabo en XML, la representación de datos no posee ningún formato específico ya que XML es texto-simple; aunado a esto XMLRPC/SOAP ha sido diseñado alrededor de HTTP, protocolo utilizado en Internet, esto no sólo facilita el problema de acceso a "Firewalls", sino que abre la puerta a la posibilidad de acceder métodos remotos en máquinas de cualquier empresa; en el proceso automatizando realiza intercambios de información de una manera transparente.

XMLRPC fue el primer mecanismo que surgió para invocar procedimientos remotos vía XML, ofrece una manera muy sencilla de invocar operaciones en sistemas heterogéneos a través de una estructura simple; SOAP es una implementación más robusta para llevar a cabo una intercomunicación en XML, a diferencia de XMLRPC a SOAP se le han integrado diversos mecanismos que le permiten operar en ambientes distribuidos más complejos tales como: un lenguaje neutro para su descripción como WSDL, y directorios distribuidos para su ubicación como UDDI, o ebXML.

En XMLRPC siempre se habla en términos de cliente/servidor, existe un sistema que realiza la solicitud denominado "el cliente" y otro que la atiende denominado "el servidor", y teniendo como elemento clave en ambos puntos: el parser XML.

Aunque es posible diseñar desde cero cualquier tipo de cliente y/o servidor para emplear XML, hoy en día ya existen diversas implementaciones para diversos ambientes y lenguajes, es a través de estas implementaciones que se logran ahorrar diversas labores como configuraciones de parser, cuestiones de seguridad, integración a servidores de páginas y otros detalles secundarios, utilizando estas estructuras "Frameworks" se concentra en los procedimientos específicos y no en detalles comunes o secundarios que siempre son utilizados en XMLRPC.

1.4.4 SOAP.

SOAP (Simple Object Access Protocol) es un protocolo ligero basado en XML, para el intercambio de información en un ambiente descentralizado y distribuido, que opera en un contexto más amplio, el de los servicios Web, unido con el estándar UDDI para proporcionar servicios de registro y mensajería entre los negocios. [7]

El protocolo SOAP es la evolución del protocolo XML-RPC que permite hacer llamadas a procedimientos remotos a través de Internet usando XML como lenguaje común y HTTP como protocolo de transporte, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web. [7]

Dado que SOAP utiliza XML para codificar mensajes, es relativamente sencillo procesar los mensajes en cada paso del proceso. Además, la facilidad de depuración de mensajes SOAP permite la convergencia rápida de las diversas implementaciones de SOAP, la cual es importante en la interoperabilidad a gran escala. [7]

Si bien SOAP puede usar otros protocolos de transporte, el uso de HTTP permite ampliar el rango de los puntos de acceso, al pasar fácilmente a través de firewalls, esto es porque se comunica en la mayoría de los casos vía los puertos 80 y 443 que ya están abiertos en la mayoría de los firewalls en el mercado y por lo tanto es importante garantizar la seguridad de estos mensajes cuando entran y salen de las redes. [16]

SOAP puede intercambiar una variedad de datos que pueden estar dentro de un XML, por lo tanto es importante garantizar los encabezados SOAP de estos mensajes para evitar definiciones erróneas. También es importante autenticar y garantizar la integridad de los mensajes SOAP que están

solicitando servicios en diferentes redes y plataformas. La especificación SOAP indica cómo se deben codificar los mensajes que circularán entre las dos aplicaciones. [7]

La especificación SOAP define dos modelos de mensajes:

- Un mensaje que se enviará desde la aplicación cliente a la aplicación servidor, solicitando la ejecución de un método al que se pasan una serie de parámetros. Anexo 5
- Un mensaje que se enviará desde la aplicación servidor a la aplicación cliente, y que contendrá datos XML con los resultados de la ejecución del método solicitado. Anexo 6

Arquitectura de SOAP.

Esta diseñado para llevar acabo intercambios de información en XML para sistemas altamente distribuidos, en especifico: "Internet". Uno de los conceptos integrados a SOAP que no existe en XMLRPC es el uso de un lenguaje neutro WSDL (descendiente de XML) para describir las funciones/métodos residentes en "el servidor", esto tiene una ventaja muy evidente: [7]

- Al utilizar un lenguaje neutro WSDL para describir las funcionalidades de "el servidor", éste funciona como un contrato al que se deben apegar los distintos "clientes"; lo anterior facilita que puedan ser escritos "clientes" en diversos lenguajes a partir de este contrato. [7] Anexo 7

La idea es la siguiente: quien publica un servicio, crea también estas páginas. Quien quiera llamar el servicio, puede utilizar estas páginas como "documentación" de la llamada y también utilizarlas antes de llamar las funciones para verificar si cambió algo. [7]

Implementaciones de SOAP.

Para facilitar la creación y el formateo de los mensajes SOAP que deben intercambiar las aplicaciones Web, los programadores disponen de distintas utilidades y aplicaciones. Estas herramientas para desarrollar aplicaciones SOAP incluyen generadores de WSDL para distintos lenguajes, servidores UDDI y otros paquetes más; algunas de estas herramientas son:

- Toolkit IBM Web Services para el lenguaje Java.
- Toolkit Microsoft SOAP, para "COM/ VBasic/ .Net".
- Soap::Lite para el lenguaje Perl.

- Web Services Pack de Sun para el lenguaje Java.
- Apache SOAP para el lenguaje Java.
- Apache Axis para Java.
- WASP para los lenguajes C++ y Java.
- librería "nuSOAP" en PHP.

Independientemente de cómo se haga la solicitud, las respuestas siempre son en XML, en el se describe perfectamente los datos en tiempo de ejecución y evita los problemas ocasionados por cambios inadvertidos en las funciones, ya que los objetos llamados tienen la posibilidad de validar siempre los argumentos de las funciones, haciendo que el protocolo sea muy sólido. [7]

1.4.5 WSDL

A partir de WSDL (Web-Services Description Language) se describen las funcionalidades del servicio web, en XMLRPC es necesario conocer de antemano que funciones/métodos residen en "el servidor" y no solo esto, sino que además se requiere conocer el lenguaje en el que está escrito "el servidor". WSDL logra aislar el lenguaje específico del servicio web. Logrando que se puedan distribuir los detalles del servicio web en un lenguaje *neutro*, permitiendo que la definición (Cliente) sea implementada en distintos lenguajes y ambientes tales como: *C++*, *Perl* o *VBASIC/.NET*.

De esta forma es posible publicar los servicios web en UDDI, permitiendo que estos sean descubiertos en un directorio conocido y centralizado.

Este mecanismo facilita el intercambio comercial por un medio electrónico, algo que resulta difícil a través de XMLRPC ya que no existe una metodología para *descubrir* servicios en una red como Internet; desde luego un desarrollo con SOAP es sumamente más complejo que con XMLRPC, esto se debe principalmente a que se debe interactuar con tecnologías adicionales como WSDL y UDDI, pero a su vez ofrece mayor flexibilidad y potencial a escala.

La secuencia de pasos sería de esta forma: Un desarrollador puede utilizar la descripción para diseñar una aplicación que se conecte a un servicio Web. Estos servicios web deben ser encontrados en la red, y es ahí donde juega UDDI un papel importante donde se pueden catalogar y buscar los servicios Web y su correspondiente descripción.

1.4.6 Seguridad.

La necesidad de garantizar la integridad, la confidencialidad y la autenticidad de los datos que fluyen a través de la Web se ha convertido en un requisito esencial. Por este motivo el área de seguridad crece rápidamente, pero hay muchas dificultades a la hora de manejar datos con estructuras jerárquicas y con subgrupos de datos con diferentes requisitos en lo que se refiere a confidencialidad, derechos de acceso o integridad, UDDI utiliza diversos mecanismos para asegurar la integridad de los servicios web y de la información que maneja, estos serán explicados en este epígrafe.

WS-Security

Con SOAP, cada mensaje simple que se intercambia realiza múltiples saltos y es rutado a través de numerosos puntos antes de que alcance su destino final. Es por ello que los Servicios Web necesitan tecnologías que protejan los mensajes desde el principio hasta el final. [8]

WS-Security define un elemento de encabezado SOAP para transportar datos relacionados con la seguridad. Si se utiliza XML Signature, este encabezado puede contener la información definida por XML Signature que transmite el modo de firma del mensaje, la clave utilizada y el valor de firma resultante. Asimismo, si un elemento del mensaje se encuentra cifrado, la información de cifrado como, por ejemplo, la que transmite XML Encryption, se puede incluir en el encabezado de WS-Security. Ésta no especifica el formato de la firma ni el cifrado, sino que determina el modo en que se puede incorporar la información de seguridad que aportan otras especificaciones en un mensaje SOAP. [9]

Con WS-Security se pretende transportar estos conceptos sobre identificación y autorización al ámbito de la mensajería SOAP. Para poder aplicar medidas que resulten significativas en un mensaje SOAP, éste debe incluir información donde entren en juego las siguientes operaciones:

- Identificar la entidad o entidades asociadas con el mensaje.
- Comprobar que las entidades disponen de las pertenencias a grupos correctas.
- Comprobar que las entidades constan del conjunto adecuado de derechos de acceso.
- Comprobar que el mensaje no se ha modificado.

Es conveniente disponer de un mecanismo que pueda impedir a personas no autorizadas, visualizar la información, WS-Security permite aplicar estos conceptos a los mensajes SOAP. Mediante los token de seguridad para identificar al remitente y comprobar sus derechos, los mensajes pueden transmitir la siguiente información: [9]

- Identidad del autor de la llamada
- Pertenencia al grupo
- Afirmaciones de derechos

XML Signature

Cuando se firma un mensaje, es muy difícil que éste se pueda ver alterado. La firma no protege el contenido del mensaje de accesos no autorizados. Con la presencia de la firma, el receptor del mensaje SOAP puede saber qué elementos no se han modificado durante el envío. Siempre que se pueda, se debe utilizar XML Signature, debido a que administra una serie de elementos difíciles que se pueden averiguar. WS-Security simplemente explica cómo utilizar la firma para comprobar que el mensaje no se ha modificado. [9]

XML Encryption y XML Signature, están preparados para manejar situaciones en las que partes de un mismo documento necesitan un tratamiento diferente, como ocurren en documentos con diferentes secciones cuyo contenido puede ser visto por unos usuarios pero no por otros. En estos casos la encriptación juega un papel muy importante ya que es lo que va a confirmar la integridad del texto. Por otro lado, las firmas digitales permiten la autenticación del remitente. Otro problema añadido surge cuando diferentes personas firman digitalmente un mismo documento XML o cuando es necesario hacerlo conjuntamente codificando ciertas partes de ese documento.

XML Signature asegura la integridad de partes de documentos XML transportados. También proporciona la autenticación de mensajes y/o servicios de autenticación de firma para datos de cualquier tipo, tanto si se encuentra en el XML que incluye la firma o en cualquier otra parte. Puede aplicarse a cualquier contenido digital (objeto de datos), incluyendo XML. Lo que hace principalmente XML Signature es asociar claves con los datos de consulta. XML Signature representa un sistema que a través de una firma digital permite ofrecer autenticidad de los datos. Con la firma digital se confirma

la identidad del emisor, la autenticidad del mensaje y su integridad, sin olvidar que los mensajes no serán repudiados.

XML Encryption.

Existen situaciones en las que no es suficiente con comprobar la identidad del remitente y demostrar que el contenido del mensaje no se ha visto alterado. Es necesario que los datos se cifraran de tal modo que sólo la persona a la que va destinado el mensaje pudiera consultarlos. Cualquier persona que observe el intercambio de conexión permanecería ajena respecto al contenido del mensaje. Tal y como sucede con la firma de mensajes, la especificación WS-Security toma las medidas adecuadas, adopta un estándar ya existente y, asimismo, realiza la tarea de cifrado. Es decir, se incorporó XML Encryption. [9]

XML Encryption es una Recomendación del Consorcio Web que especifica un proceso para encriptar datos (no únicamente documentos XML) y representar esa información encriptada a su vez en XML para que viaje por los medios de transmisión.

XML Encryption establece que para encriptar un elemento (o un contenido textual de un elemento), este va a ser sustituido por otro elemento establecido por la especificación que contendrá el dato encriptado y la información necesaria para el procesamiento, como puede ser la identificación del algoritmo utilizado o datos sobre la clave empleada.

Al perfilar como mecanismo de comunicación entre las plataformas el XML, como forma de comunicarse el protocolo SOAP, se da margen a la siguiente pregunta; ¿cómo sabrán con quien comunicarse y dónde encontrar otros negocios? La respuesta es UDDI. Esta tecnología será el medio de localizar los servicios Web, que están tomando cada vez más importancia por su capacidad de ocultar la complejidad de los sistemas finales y permitir una fácil comunicación. Para UDDI será una tarea más difícil conforme crece el número de servicios disponibles, y hacia la solución de esta tarea va encaminada el desarrollo creciente de UDDI.

1.5 La arquitectura de informatización de la UCI y la implementación de UDDI.

Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, la dirección de informatización definió las tendencias y tecnologías actuales posibles a emplear, descritas a continuación.

1.5.1 Las aplicaciones Web.

Las aplicaciones Web son una especialización y concreción de las aplicaciones cliente-servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (HTTP) son estándar, y no han de ser creados por el desarrollador.

La creciente popularidad de las aplicaciones Web se debe a sus múltiples ventajas, entre las cuales podemos citar:

- **Multiplataforma:** Con un solo programa, un único ejecutable, nuestras aplicaciones pueden ser utilizada a través de múltiples plataformas, tanto de hardware como de software.
- **Actualización instantánea:** Debido que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.
- **Suave curva de aprendizaje:** Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
- **Fácil de integrar con otros sistemas:** Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- **Acceso móvil:** El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

El desarrollo de aplicaciones Web está siendo utilizado en muchas organizaciones, ésta situación va ir creciendo indefinidamente. Es por ello que día a día se requieran más programadores capacitados para desarrollos basados en el World Wide Web (WWW). Es necesario para darle solución al problema crear una aplicación Web que permita al usuario buscar e insertar información acerca de los servicios web.

1.5.2 Modelo Cliente Servidor.

Se dice que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes: [10]

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas de la arquitectura cliente-servidor:

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

La arquitectura de software de tres capas emergió en la década de los noventas para solventar las limitaciones de la arquitectura de dos capas. La tercera capa (capa de servicios) se localiza entre la interfaz de usuarios (cliente) y el administrador de datos (servidor). Esta capa intermedia provee de servicios para la administración de procesos (tal como desarrollo, monitoreo y alimentación de procesos) que son compartidos por múltiples aplicaciones. [10] *Ver Anexo 1*

El servidor de la capa intermedia (también conocido como servidor de aplicaciones) centraliza la lógica de las aplicaciones, haciendo que la administración de cambios sea más sencilla. En arquitecturas más simples, cualquier cambio en la lógica, implica reescribir todas las aplicaciones que dependan de ésta.

1.5.3 PHP (PHP: Hypertext Preprocessor).

El PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La dirección de informatización ha determinado el uso de PHP 5 como herramienta para implementar la propuesta, utilizando el motor Zend Engine II.

Una de sus características más potentes es su soporte para gran cantidad de bases de datos, entre las que se pueden mencionar InterBase, mSQL, MySQL, Oracle, entre otras , y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.

PHP también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (Portable Document Format) hasta analizar código XML y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+. [10]

Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas. Es multiplataforma, funciona tanto para Unix con Apache como para Windows con Microsoft Internet Information Server.

Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP (Active Server Pages).

Afortunadamente, lo nuevo de PHP 5 mejora muchas áreas en el lenguaje y su ejecución, como por ejemplo: [11]

- Programación orientada a objetos (OOP).
- MySQL.
- XML.
- Integración nativa con el Zend Engine.

1.5.4 Servidor Web Apache.

Se decidió utilizar el servidor Apache por ser un software de código abierto que funciona sobre cualquier plataforma. Tiene capacidad para servir páginas tanto de contenido estático, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información, es muy potente y altamente configurable. [10]

El servidor Apache es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías: [10]

- Módulos Base: Módulo con las funciones básicas del Apache.
- Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. [10]

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. [10]

1.5.5 AJAX.

También se utilizará AJAX, (**A**synchronous **J**avaScript **A**nd **X**ML), para crear aplicaciones interactivas. Ejecutándose en el cliente, es decir, en el navegador del usuario, y mantener comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la velocidad de interacción en la misma. [12]

AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente y que se muestra a continuación: [12]

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto IFrame en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML.

1.5.5.1 Ventajas del AJAX:

- Interactividad: Puede ser usado para multitud de tareas como actualizar o eliminar registros, expandir formularios web, devolver peticiones simples de búsqueda, o editar árboles de categorías; todo sin tener la necesidad de recargar toda la página de HTML cada vez que se realiza un cambio. Generalmente solo requiere enviar pequeñas peticiones al servidor, y se devuelven respuestas relativamente cortas. Esto permite el desarrollo de aplicaciones interactivas con más interfaces de usuario más responsivas gracias al uso de las técnicas DHTML. [12]
- Portabilidad: Las aplicaciones construidas con AJAX utilizan características bien documentadas presentes en todos los navegadores importantes en la mayoría de las plataformas existentes. Aunque esta situación podría cambiar en el futuro, en este momento, los usos de AJAX son efectivos entre plataformas. Mientras que la plataforma de AJAX está más restringida que la plataforma de Java, las aplicaciones actuales de AJAX llenan con eficacia la parte de los Java Applets: ampliar el navegador con mini-aplicaciones ligeras. [12]

1.5.6 Patrones de Diseño.

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Los patrones capturan la experiencia existente y probada para promover buenas prácticas. [10]

Los Patrones de Diseño (Design Patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Estos se dividen en tres grandes categorías:

- Patrones Creacionales: Solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.
- Patrones Estructurales: Solucionan problemas de composición (agregación) de clases y objetos.
- Patrones de Comportamiento: Soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

La dirección de informatización estableció el uso del patrón Modelo Vista Controlador para las aplicaciones Web.

1.5.7 Modelo Vista Controlador (MVC).

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

El **Modelo** es todo acceso a datos, y las funciones que llevan se llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo modelo. [10]

La **Vista**, en una aplicación web, es el HTML y lo necesario para convertir datos en HTML. O sea muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código.

El **Controlador** es lo que une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las entradas del usuario. Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.). Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. De este modo, el código que "hace algo" está perfectamente separado del código dedicado a crear HTML, lo que ayuda a evitar el spaghetti.

Un ejemplo típico del funcionamiento de este patrón se puede observar en el *Anexo 5*.

El MVC es un patrón ampliamente utilizado en múltiples plataformas y lenguajes. Algunos de sus principales beneficios son: [10]

- Menor acoplamiento.
 - Desacopla las vistas de los modelos.
 - Desacopla los modelos de la forma en que se muestran e ingresan los datos.
- Mayor cohesión.
 - Cada elemento del patrón esta altamente especializado en su tarea.
- Las vistas proveen mayor flexibilidad y agilidad.
 - Se puede crear múltiples vistas de un modelo.
 - Se puede crear, añadir, modificar y eliminar nuevas vistas dinámicamente.
 - Las vistas pueden anidarse.
 - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
 - Se puede sincronizar las vistas.

- Las vistas pueden concentrarse en diferentes aspectos del modelo.
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales.
 - Una vista para cada dispositivo que puede variar según sus capacidades.
 - Una vista para la Web y otra para aplicaciones de escritorio.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

1.5.8 Sistemas de Gestión de Base de Datos.

Un Sistema de Gestión de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. Se determinó utilizar PostgreSQL como sistema gestor de bases de datos.

1.5.8.1 PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos Objeto-Relacionales (ORDBMS) libre, liberado bajo la licencia BSD (Berkeley Software Distribution). Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle y SQLServer.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x.

- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

- Altamente Extensible: PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.
- Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL (Tool Command Language) como lenguaje procedural embebido.
- MVCC: Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Los bloqueos son provocados por usuarios que están escribiendo en la base de datos, el lector está bloqueado por los escritores que están actualizando registros.
- Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Las principales mejoras en PostgreSQL incluyen: [10]

- Los bloqueos de tabla han sido sustituidos por el MVCC, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.
- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han añadido características adicionales que cumplen el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadenas literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

1.5.9 Proceso de Desarrollo.

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo. Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de automatización, se ha escogido el Proceso Unificado de Desarrollo de Software (RUP). [13]

RUP es un proceso de desarrollo de software, o sea, conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de empresas. [13]

Es un proceso basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en que está dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, los cuales se muestran en el *Anexo 6*, que parten de los casos de uso; centrado en la arquitectura y es iterativo e incremental. [13]

Está acompañado de una herramienta muy buena que soporta cada uno de los procesos que necesitamos: Rational Rose Enterprise Edition 2003. Además cubre el ciclo de vida de desarrollo de una empresa y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software. [13]

Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Sus creadores pretendieron con este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar. [13]

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el

proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas. [13]

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. [14]

Ver en el anexo 4 los elementos, relaciones y diagramas que conforman el UML. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará.

De forma general las principales características son: [13]

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas de la empresa.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que dan asistencia a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los sistemas. La herramienta fue desarrollada por los creadores de UML, utilizando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de una empresa. [15]

1.5.10 Herramientas utilizadas.

1.5.10.1 Diseño de interfaz.

Macromedia Dreamweaver MX es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la

más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa y utilizar casi todos los recursos de la Web.

Este editor de HTML profesional para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, o sea, crear páginas rápidamente sin escribir una línea de código, así como también la codificación manual. Dreamweaver ayuda además a construir aplicaciones Web dinámicas apoyadas en bases de datos.

Es completamente personalizable. Se pueden crear objetos y comandos propios, modificar los accesos directos de teclado, e incluso escribir código JavaScript para extender las capacidades del Dreamweaver con nuevos comportamientos. Soporta varias tecnologías del servidor para la construcción de aplicaciones Web, tales como: Macromedia ColdFusion, Microsoft ASP, Microsoft ASP.NET, Sun JavaServer Pages (JSP) y PHP.

1.5.10.2 Zend Studio.

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica las empresas complejas.

1.5.10.3 Adobe Photoshop.

Adobe Photoshop CS para el tratamiento de los gráficos. Es una herramienta muy poderosa para crear cualquier tipo de gráficos, su integración con Adobe ImageReady hacen que crear complicados gráficos para la Web sea una tarea muy fácil.

1.5.10.4 Rational Rose.

Existen herramientas CASE de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los empresas, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose y es la que se utiliza en la modelación de este empresa.

Rational Rose cubre todo el ciclo de vida de un empresa: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del empresa.

Se decidió que se utilizaría el Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

1.5.10.5 PgAdmin

Es desarrollado por una comunidad de los expertos de PostgreSQL de varias partes del mundo y está disponible en más de una docena de idiomas. Se trata de una herramienta para la administración de bases de datos PostgreSQL. Su uso se puede extender hacia las plataformas de Linux, de FreeBSD, de Solaris, del Mac OSX y de Windows.

1.6 Conclusiones.

En este capítulo se exponen las condiciones y problemas que rodean el objeto de estudio; y a través de los conceptos y definiciones planteadas. Se evidencia la necesidad de implementar un software que permita la interoperabilidad, organización y publicación de los servicios Web en el contexto de la arquitectura SOA. Para desarrollar el sistema se hace uso de la tecnología para la programación de páginas dinámicas el lenguaje PHP5 y con soporte de base de datos en PostgreSQL. El proceso de desarrollo es RUP, el cual está basado en la orientación a objetos y el modelamiento visual usando

UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios.

CAPÍTULO
2
Modelo del Sistema

2.1 Introducción

En este capítulo se realiza un estudio de la situación actual del problema a resolver a través de una descripción de los procesos de negocios, obteniendo como resultado de la inexistencia del negocio se necesita realizar un Modelo de Dominio, en el cual se definen conceptos que permiten realizar una correcta captura de requisitos y poder construir un sistema correcto.

Además se definen los requerimientos funcionales y no funciones que debe tener la propuesta de sistema, lo que posibilita que se puedan identificar actores y casos de uso y construir un Diagrama de Caso de Usos del Sistema.

2.2 Descripción de los proceso del negocio propuestos.

Para describir los procesos del negocio que se relacionan con el campo de acción de este trabajo, es necesario conocer que tipo de topología de UDDI aplicar en las empresas, para luego centrar la atención en los principales procesos a realizar.

Con creciente desarrollo de la informática en Cuba se ha incrementado el número de profesionales y con ello la producción de software para el comercio y la informatización de la sociedad cubana, esto trae consigo que se aumente el número de aplicaciones y servicios web en el país.

Se propone que se implante una aplicación UDDI con una topología de tipo Nodo Privado UDDI, este registro si se desea no será accesible desde Internet ni por ninguna otra institución, por lo que sus datos no se replicarán con ningún otro servidor UDDI. Este tipo de topología trae consigo que se puedan definir las propias políticas de seguridad y acceso. Lo cual permite que la propuesta del trabajo se convierta en una excelente herramienta para la publicación, búsqueda y categorización de servicios web que serán re usados por parte de los desarrolladores de aplicaciones de las empresas haciendo que la infraestructura entre los sistemas de una organización se convierta en un ambiente ínter operable. [1]

Proceso de Búsquedas.

Este proceso tiene como objetivo mostrar la información necesaria para desarrollar un sistema informático, con la idea de que cualquier persona que trabaje en el entorno organizacional pueda utilizar el Nodo Privado UDDI para consultar información de una empresa, un servicio o de las especificaciones técnicas del servicio.

La información se distribuye en tres partes: para acceder a la información de la empresa proveedora de Servicios Web, se recurre a la *búsqueda de empresa*; para acceder a la referencia del servicio se recurren a la *búsqueda de servicio* y para acceder a las especificaciones técnicas de un servicio se recurren a la *búsqueda de especificaciones técnicas*.

A través del sistema, se trataría de agrupar toda información de una empresa, un servicio o de las especificaciones técnicas del servicio en un catálogo, de manera tal que una vez que se solicite una de estas opciones el sistema busca la información que es solicitada, siempre que esta exista, y en caso posible visualizar el contenido del mismo.

Proceso de Registro

Este proceso tiene como objetivo registrar toda la información que se puede necesitar para el desarrollo del sistema, para el cual es necesario que el especialista que trabaje en el entorno organizacional se registre en el Nodo Privado UDDI y se convierta en un proveedor, lo que permite que pueda publicar su empresa, sus servicios y especificaciones técnicas.

La información se distribuye en tres partes: para guardar información de la empresa proveedora del Servicio Web, se recurre al *registro de empresa*; para guardar el servicio se recurren al *registro del servicio* y para guardar las especificaciones técnicas de un servicio el proveedor debe ir al *registro de especificaciones técnicas*.

Proceso de Administración

Este proceso tiene como objetivo controlar toda la información existente y administrar las funcionalidades del sistema, con la idea de que el administrador del Nodo Privado UDDI sea el encargado de garantizar que el Nodo cumpla con todos los requisitos establecidos.

Con la propuesta de este trabajo, se trataría de que todos los usuarios que tengan acceso puedan incorporar al catálogo de servicio información de su empresa, servicio o de la especificación técnica.

2.3 Modelo del Dominio

Teniendo en cuenta las descripciones de los procesos en el epígrafe anterior, se concluye que para dar solución al negocio que se está estudiando es necesario realizar un Modelo de Dominio, debido a la inexistencia de este en el entorno organizacional.

El Modelo del dominio es el encargado de capturar los tipos más importantes de objetos que existen o los eventos que suceden en el entorno de la organización, permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los clientes, usuarios, y desarrolladores a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. El primer paso es definir los principales conceptos.

2.3.1 Conceptos:

- Se le denominará **cliente**, a cualquier persona que trabaje en el entorno organizacional, sin importar su categoría o lugar de residencia.
- Se le denominará **proveedor**, a aquel usuario que tenga permiso o acceso para gestionar (registrar, eliminar, modificar) uno o varios datos dentro de la UDDI.
- Se le denominará **administrador**, a aquel usuario que tenga permiso o acceso para administrar (gestionar, buscar, administrar usuario) todos los datos dentro de la UDDI.
- Se le denominara **Entono organizacional**, a todas la empresas que pertenezcan a la asociación UDDI,
- Se denominará **UDDI**, a todo el sistema que permita registrar, buscar información de forma estructurada sobre los servicios Web y las especificaciones técnicas de estos, que realiza cada empresa.
- Se denominará **información UDDI**, a todo término que brinde información acerca de una empresa, de sus servicios Web y las especificaciones técnicas de estos.

2.3.2 Diagrama del modelo de dominio

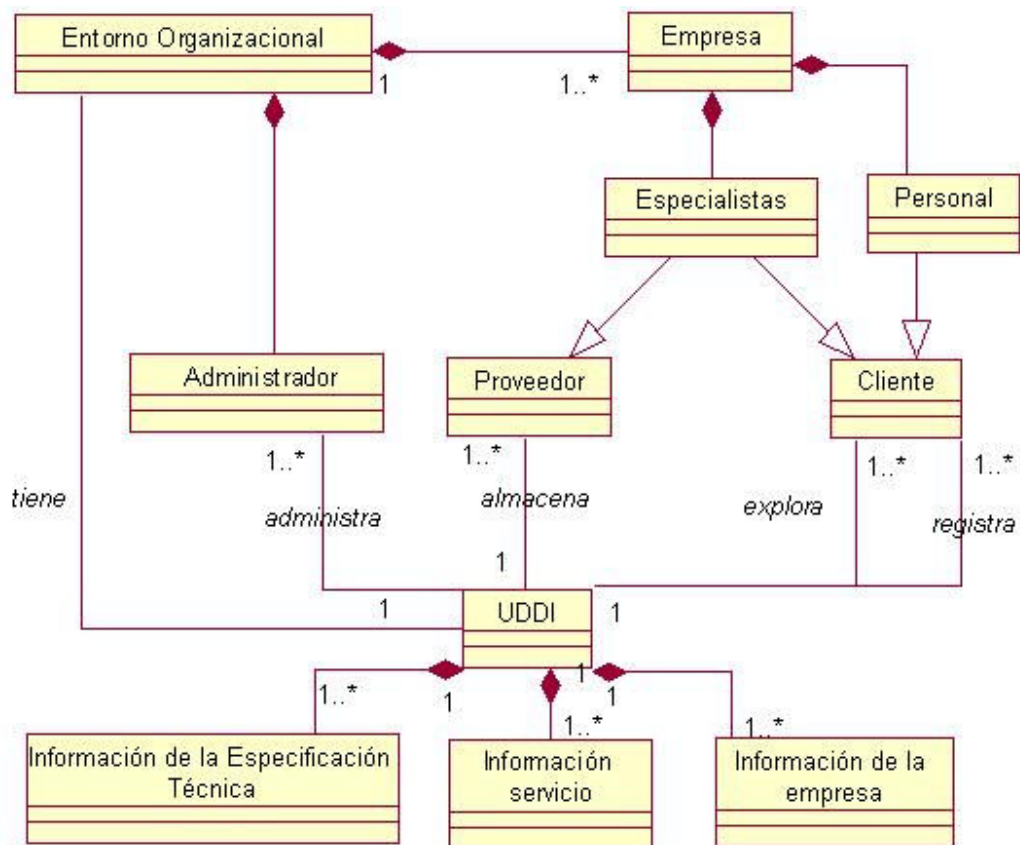


Figura 2.1 Modelo de Dominio

2.4 Requerimientos

2.4.1 Requerimiento funcional

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del software.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

1. Autenticar usuario
 - 1.1. Comparar Usuario y contraseña con los usuarios del sistema.
 - 1.2. Modificar contraseña.
 - 1.3. Asignar privilegios.
2. Administrar
 - 2.1. Crear categoría de empresa.
 - 2.2. Crear categoría de servicio.
 - 2.3. Crear categoría de referencia.
 - 2.4. Crear categoría de Especificaciones técnicas.
 - 2.5. Crear identificadores de empresa.
 - 2.6. Crear identificadores de Especificaciones técnicas
3. Administrar usuarios del sistema.
 - 3.1. Registrar usuario.
 - 3.2. Generar cuenta de usuario.
 - 3.3. Eliminar usuario.
 - 3.4. Bloquear usuario.
 - 3.5. Desbloquear usuario.
 - 3.6. Buscar usuario.
 - 3.7. Mostrar lista de usuarios.
 - 3.8. Mostrar lista de solicitudes de usuarios.
 - 3.9. Mostrar solicitud de usuarios.

- 3.10. Revisar solicitud de usuarios.
- 3.11. Clasificar usuario.
- 3.12. Ver historial de usuario.
- 3.13. Controlar acceso a la información.
 - 3.13.1. Definir niveles de acceso a la información.
- 4. Solicitud de cuentas de usuarios.
- 5. Buscar
 - 5.1. Buscar servicios.
 - 5.2. Buscar empresa.
 - 5.3. Buscar Especificación técnica.
 - 5.4. Mostrar toda la información del servicio.
 - 5.5. Mostrar toda la información de la empresa.
 - 5.6. Mostrar toda la información de la Especificación técnica.
 - 5.7. Enviar sugerencia o problema.
- 6. Crear empresa.
- 7. Modificar empresa.
- 8. Eliminar empresa.
- 9. Adicionar contacto.
- 10. Modificar contacto.
- 11. Eliminar contacto.
- 12. Crear servicio.
- 13. Modificar servicio.
- 14. Eliminar servicio.
- 15. Crear Punto de acceso.
- 16. Modificar Punto de acceso
- 17. Crear Especificación técnica.
- 18. Modificar Especificación técnica.
- 19. Eliminar Especificación técnica.
- 20. Configurar nodo.
- 21. Publicar

- 21.1. Publicar una empresa.
- 21.2. Publicar servicio.
- 21.3. Publicar especificaciones técnicas.
- 21.4. Denegar servicio.
- 21.5. Denegar Especificación técnica.
- 21.6. Denegar empresa.
22. Controlar modificaciones.
23. Generar XML.
24. Comprobar API.
25. Enviar resultado.

2.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto.

A continuación se muestran los requerimientos no funcionales:

- **Apariencia o interfaz externa**

La interfaz no contendrá muchas imágenes para no demorar las respuestas al usuario ya que en este tipo de sistemas la velocidad del sistema es una característica imprescindible.

El diseño de la interfaz será sencillo y claro de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Será formal, serio y con una navegación sugerente, todo esto teniendo en cuenta el fin con el que se desarrolla la aplicación.

- **Usabilidad**

El sistema ha sido diseñado para que sea fácil de manipular; permite a los usuarios tener control en todo momento del sitio web, evitando para eso la sobrecarga de información y para cada proceso una muestra a través de mensaje de las acciones hechas por los usuarios. Los colores usados permiten el acceso a usuarios con problemas de distinción de color, se reduce el tiempo de atención, además, requiriendo un mínimo proceso de aprendizaje. Esta diseñado para que lo usen los distribuidores de

servicios web, por lo que instalar el sistema trae consigo una mayor rapidez de trabajo y por consiguiente un ahorro de materiales y personal.

- **Rendimiento**

La disponibilidad de trabajo en red contra el servidor será constante. Se garantizará que la respuesta a solicitudes de los usuarios del sistema sea en un período de tiempo breve (de segundos) para evitar la acumulación de trabajo por parte de los responsables y público en los puntos de admisión. El sistema deberá de ser lo más estable y confiable posible.

- **Soporte**

En un futuro cuando el software llegue a la etapa de implementación, el producto recibirá mantenimiento ante cualquier fallo que ocurra. El sistema es de fácil instalación.

- **Ayuda y documentación en línea**

En un futuro cuando el software llegue a la etapa de implementación, el producto brindará a los usuarios una buena ayuda en línea de modo que si el usuario presenta algún problema cuando vaya a realizar una búsqueda específica, pueda acudir al mismo, así como una documentación apropiada para el mejor trabajo con el mismo.

- **Software**

Para el funcionamiento del sistema en el servidor será necesario el SO Windows 98 o superior, Linux o Unix, en sus versiones de SO servidores. Para el funcionamiento del sistema en las terminales cliente será necesario el SO Windows 98 o superior, Linux o Unix.

- **Hardware**

Se necesitan como requerimientos mínimos una PC con procesador Pentium II o superior.

- **Portabilidad**

El producto es usado bajo los SO Windows, Linux y Unix. El producto corre sobre una plataforma Web, codificada en "PHP5" y sus sistemas de bases de datos en PostgreSQL.

- **Seguridad**

El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios al sitio, de identificar al administrador antes de que pueda realizar cualquier acción sobre el sistema. Garantiza que la información sea vista únicamente por quien tiene derecho a verla.

Existe un primer nivel o nivel básico donde están las funciones asociadas al usuario general o común donde está la información pública, que requiere poca responsabilidad. El segundo nivel está compuesto por información más confidencial por lo tanto solo el personal autorizado puede verla. El tercer nivel está conformado con las funciones administrativas del sitio de mayor complejidad que pueden destruir información relacionada a las entidades del sistema por lo tanto es el nivel de mayor complejidad.

Se usan mecanismos de encriptación (Base64) de los datos que por cuestiones de seguridad no deben viajar al servidor en texto claro, como es el caso de las contraseñas.

Se hacen validaciones de la información tanto en el cliente como en el servidor, no obstante los usuarios acceden de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

EL intercambio de mensajes entre los clientes y la aplicación se realiza con el protocolo SOAP, un protocolo de mensaje liviano basado en XML, usado para codificar los mensajes de Servicio Web antes de enviarlos por la red.

- **Confidencialidad**

Toda la información está protegida del acceso no autorizado, los administradores de sistema son los únicos que podrán transformar la información, los operadores solo podrán ver los listados de información.

- **Disponibilidad**

Se garantizará a los usuarios del sistema el acceso a la información solicitada en todo momento siempre que tenga permiso para ello.

- **Políticos-Culturales**

Toda modificación al funcionamiento establecido en los requerimientos será realizada por la Dirección del Puesto Mando Informatización conjuntamente con el personal enfrente del sistema.

- **Restricciones en el diseño y la implementación**

Es una aplicación Web desarrollada con la tecnología para creación de páginas Web dinámicas PHP5 y base de datos en PostgreSQL.

- **Legales**

El sistema se basa en un estándar que se rige por normas internacionales y cumple con las normas y leyes establecidas en nuestro país. La plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GNU/GPL.

- **Confiabilidad**

La herramienta de implementación a utilizar tendrá soporte para recuperación ante fallos y errores.

- **Restricciones**

Se utiliza UML para lograr una mejor documentación del sistema y como herramienta de apoyo Rational Rose. Se utiliza como lenguaje de programación PHP5 y el gestor de base de datos PostgreSQL.

2.5 Identificación de actores.

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información. En este caso los actores que interactúan con el sistema se definen a continuación.

Actor	Descripción
Cliente	Individuo que busca cierto Servicio Web en el Nodo Privado UDDI, y utiliza los servicios que están contenidos en el sistema.
Sistema externo	Sistema que busca cierto Servicio Web en el Nodo Privado UDDI, y utiliza los servicios que están contenidos en el sistema.
Proveedor	Individuo que busca y registra en el Nodo Privado UDDI los servicios que se realizaron en su empresa.

Editor	Individuo que puede buscar, publicar, y configurar datos de los servicios del Nodo Privado UDDI. Además puede ver y modificar cualquier dato almacenado de los servicios del Nodo Privado UDDI.
Administrador	Individuo que puede buscar, publicar, coordinar, y administrar los servicios del Nodo Privado UDDI. Además puede administrar opciones de servicio, manejar ajustes de la seguridad.

Tabla 2.1 Identificación de actores

2.6 Paquetes y sus relaciones.

Un sistema grande se debe dividir en unidades más pequeñas, de modo que pueda ser entendido por las personas que necesiten consultarlo. Dado el número de casos de uso y actores, se introducen paquetes en el modelo de casos de uso para tratar su tamaño, organizar los elementos en grupos y hacerlo más comprensibles.

Para satisfacer los objetivos propuestos al inicio de este trabajo el sistema que se propone debe estar conformado por cuatro paquetes: Administración, Gestionar Información, Usuario y Petición.

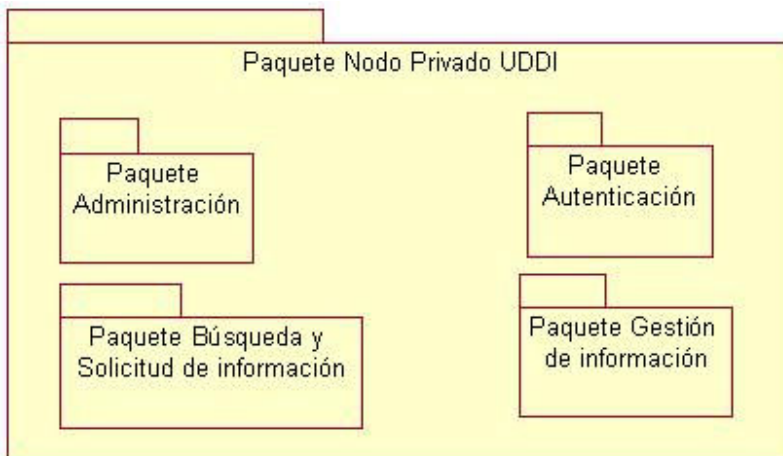


Figura 2.2 Paquetes y sus relaciones.

2.6.1 Paquetes Autenticación.

2.6.1.1 Diagrama de caso de uso.

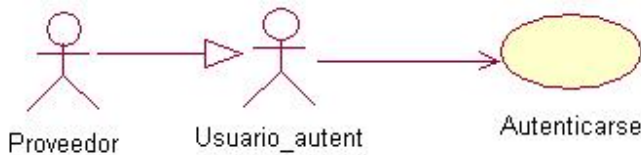


Figura 2.3 Diagrama de CU Paquete Autenticación.

2.6.1.2 Descripción de los casos de usos.

Nombre del Caso de Uso	Autenticar
Actores	Usuario_autent
Propósito	Realizar la autenticación del usuario.
Resumen	El usuario desea entrar al sistema por lo que requerirá de su correspondiente autenticación en el mismo, para lo cual debe introducir su usuario y contraseña correspondientes.
Referencias	RF 1
Precondiciones	El usuario debe tener permisos para acceder a esta parte del sistema.
Poscondiciones	<ol style="list-style-type: none"> 1. Usuario acceda a la interfaz que le corresponde. 2. Información de la contraseña cambiada, registrada en la BD.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario accede a la interfaz Autenticarse.	2. El sistema muestra una interfaz con los siguientes datos: <ul style="list-style-type: none"> • Nombre de usuario. • Contraseña.
3. El usuario introduce los datos.	4. El sistema valida los datos. 5. El sistema comprueba el acceso del usuario.

	6. El sistema muestra la página de acceso del usuario.
Curso alternativo	
Acción 3.	
3. El usuario selecciona la opción Cambiar Contraseña.	4. El sistema muestra una interfaz con los siguientes datos: <ul style="list-style-type: none"> • Nombre de usuario. • Contraseña vieja. • Nueva contraseña. • Confirmación de contraseña.
5. El usuario introduce los datos.	6. El sistema valida los datos. 7. El sistema guarda los datos. 8. El sistema envía un mensaje informando que su contraseña ha sido cambiada.
Acción 6.	El sistema muestra mensaje de error de autenticación y brinda la opción de reintentar la operación o salir.
Prioridad:	Crítico

Tabla 2.2 Descripción del CU Autenticar

2.6.2 Paquetes Búsqueda y solicitud de información.

2.6.2.1 Diagrama de caso de uso.

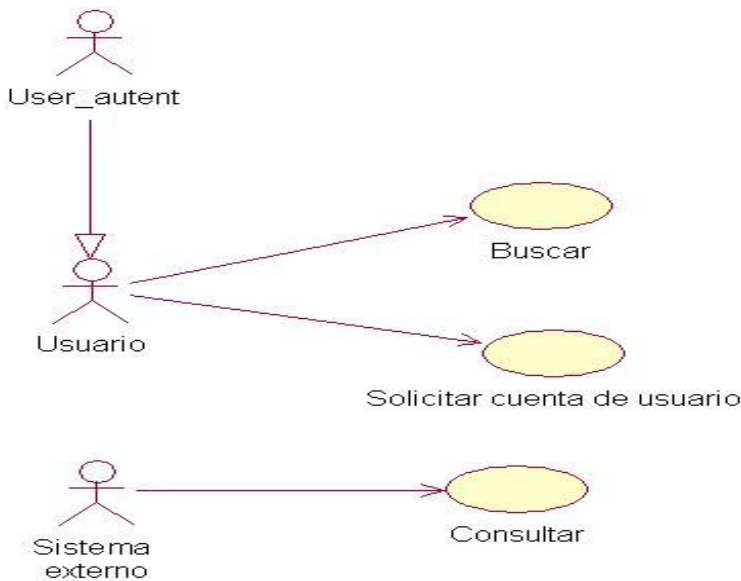


Figura 2.3 Diagrama de CU Paquete Búsqueda y solicitud de información.

2.6.2.2 Descripción de los casos de usos.

Nombre del Caso de Uso	Buscar
Actores	Cliente.
Propósito	Permitir buscar empresas, servicios o Especificaciones técnicas.
Resumen	En este caso de uso el usuario podrá buscar un servicio, una empresa o un Especificación técnica por el nombre o categoría.
Referencias	RF 3
Precondiciones	El cliente debe tener permisos para acceder a esta parte del sistema.
Poscondiciones	<ol style="list-style-type: none"> 1. Información de la empresa buscada, mostrada al usuario. 2. Información del servicio buscada, mostrada al usuario. 3. Información de la especificación técnica buscada,

	<p>mostrada al usuario.</p> <p>4. Información del correo electrónico, enviada al administrador.</p>
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
<p>1. El cliente selecciona una de las opciones de la interfaz</p> <p>Buscar</p>	<p>2. El sistema ejecuta alguna de las siguientes acciones:</p> <p>a) Si decide buscar una empresa, ir a la sección “Buscar empresa”.</p> <p>b) Si decide buscar una empresa, ir a la sección “Buscar servicio”.</p> <p>c) Si decide buscar una empresa, ir a la sección “Buscar Especificación técnica”.</p>
Sección “Buscar empresa”.	
	<p>3. El sistema muestra una interfaz con los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • Nombre de la empresa. • Categoría de empresa. • Nombre del proveedor. • Ordenar por nombre: <ul style="list-style-type: none"> ○ Ascendente. ○ Descendente.
<p>4. El cliente llena los datos.</p>	<p>5. El sistema valida los datos.</p> <p>6. El sistema busca las empresas.</p> <p>7. El sistema muestra el resultado de la búsqueda que contiene los datos:</p> <ul style="list-style-type: none"> • Nombre de la empresa. • Descripción. • Nombre del proveedor.

	<ul style="list-style-type: none"> • Correo del proveedor. • Lista de servicio.
8. El cliente selecciona la opción a realizar.	<p>9.</p> <p>a) Si selecciona el correo electrónico, ir a la sección “Enviar sugerencia o problemas”.</p> <p>b) Si selecciona mostrar detalles, ir a la sección “Mostrar empresa”.</p>
Curso Alternativo	
Acción 5	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos, y se retorna a la interfaz Buscar empresa.
Sección “Buscar servicio”.	
	<p>3. El sistema muestra una interfaz con los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • Nombre del servicio. • Categoría del servicio. • Nombre del proveedor. • Ordenar por nombre: <ul style="list-style-type: none"> ○ Ascendente. ○ Descendente.
4. El cliente llena los datos.	<p>5. El sistema valida los datos.</p> <p>6. El sistema busca los servicios.</p> <p>7. El sistema muestra el resultado de la búsqueda que contiene los datos:</p> <ul style="list-style-type: none"> • Nombre del servicio. • Descripción. • Nombre del proveedor. • Correo del proveedor. • Lista de Especificación técnica.

<p>8. El cliente selecciona la opción a realizar.</p>	<p>9.</p> <ul style="list-style-type: none"> a) Si selecciona el correo electrónico, ir a la sección “Enviar sugerencia o problemas”. b) Si selecciona mostrar detalles, ir a la sección “Mostrar servicio”.
<p>Curso Alternativo</p>	
<p>Acción 5</p>	<p>El sistema muestra un mensaje de error indicando que los campos obligatorios tienen caracteres extraños o están vacíos, y se retorna a la interfaz Buscar servicio.</p>
<p>Sección “Buscar Especificación técnica”.</p>	
	<p>3. El sistema muestra una interfaz con los siguientes parámetros de búsqueda:</p> <ul style="list-style-type: none"> • Nombre del Especificación técnica. • Categoría del servicio. • Nombre del proveedor. • Ordenar por nombre: <ul style="list-style-type: none"> ○ Ascendente. ○ Descendente.
<p>4. El cliente llena los datos.</p>	<p>5. El sistema valida los datos.</p> <p>6. El sistema busca los Especificación técnica.</p> <p>7. El sistema muestra el resultado de la búsqueda que contiene:</p> <ul style="list-style-type: none"> • Nombre del Especificación técnica. • Descripción. • Nombre del proveedor. • Correo del proveedor.
<p>8. El cliente selecciona la opción a realizar.</p>	<p>9.</p> <ul style="list-style-type: none"> a) Si selecciona el correo electrónico, ir a la sección “Enviar sugerencia o problemas”.

	b) Si selecciona mostrar detalles, ir a la sección “Mostrar especificación” .
Curso alternativo	
Acción 5	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos, y se retorna a la interfaz Buscar Especificación técnica.
Sección “Enviar sugerencia o problemas”.	
	9. El sistema muestra una interfaz que con los siguientes parámetros: <ul style="list-style-type: none"> • Dirección de correo del usuario • Asunto • Cuerpo del mensaje.
10. El cliente llena los datos 11. El cliente presiona el botón Enviar Mensaje.	12. El sistema manda el correo a la dirección de correo del administrador del Nodo Privado UDDI.
Sección “Mostrar Empresa”.	
	9. El sistema muestra una interfaz con todos los datos de la empresa: <ul style="list-style-type: none"> • Nombre de la empresa. • Descripción de la empresa. • Nombre del proveedor. • Correo del proveedor. • Categoría. • Identificador de empresa. • URL de la empresa. • Lista de Contactos. <ul style="list-style-type: none"> ○ Nombre del contacto. ○ Correo electrónico.

	<ul style="list-style-type: none"> ○ Teléfono. ○ Dirección o dirección alternativa. ● Lista de servicios. ● Lista de Especificación técnica.
Sección “Mostrar Servicio”.	
	<p>9. El sistema muestra una interfaz con todos los datos del servicio:</p> <ul style="list-style-type: none"> ● Nombre de la empresa. ● Nombre del servicio. ● Descripción del servicio. ● Nombre del proveedor. ● Correo del proveedor. ● Categoría. ● Lista de Especificaciones técnicas. ● Lista de referencia. <ul style="list-style-type: none"> ○ Tipo de Punto de Acceso. ○ URL del Punto de Acceso. ○ PC_remota. ○ Descripción.
Sección “Mostrar Especificación técnica”.	
	<p>9. El sistema muestra una interfaz con todos los datos de la Especificación técnica:</p> <ul style="list-style-type: none"> ● Nombre del servicio. ● Nombre de la Especificación técnica. ● Nombre del proveedor. ● Correo del proveedor. ● Descripción. ● Categoría. ● Identificador de la Especificación técnica.

	<ul style="list-style-type: none"> • URL de la Especificación técnica. • Tipo de URL de la Especificación técnica. • Parámetros de instancias.
Prioridad:	Crítico

Tabla 2.3 Descripción del CU Buscar.

Nombre del Caso de Uso	Consultar
Actores	Sistema externo
Propósito	Permitir que un sistema externo realice una consulta a través de las API del Nodo Privado UDDI.
Resumen	El sistema externo realice una consulta a través de las API definidas por del Nodo Privado UDDI enviándola al nodo y obteniendo una respuesta.
Referencias	RF 5.1, 5.2, 5.3, 23, 24.
Precondiciones	La consulta de la API esté correcta (bien implementada).
Poscondiciones	<ol style="list-style-type: none"> 1. Información de la empresa buscada, mostrada al usuario. 2. Información del servicio buscada, mostrada al usuario. 3. Información de la especificación técnica buscada, mostrada al usuario.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. Sistema envía una API de consulta.	<ol style="list-style-type: none"> 2. El sistema recibe la API. 3. El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> • Si la API es de búsqueda de empresa, el sistema realiza la búsqueda, genera un XML con el resultado de todos los datos de la empresa y envía el XML. • Si la API es de búsqueda de servicio, el sistema

	<p>realiza la búsqueda, genera un XML con el resultado de todos los datos del servicio. y envía el XML.</p> <ul style="list-style-type: none"> • Si la API es de búsqueda de especificación técnica, el sistema realiza la búsqueda, genera un XML con el resultado de todos los datos de la especificación técnica y envía el XML.
Prioridad:	Crítico

Tabla 2.4 Descripción del CU Consultar

Nombre del Caso de Uso	Solicitar cuenta de usuario
Actores	Cliente.
Propósito	Solicitar una cuenta de usuario que permita acceder al sistema.
Resumen	En este Caso de Uso el cliente introduce los datos que se le piden la solicitud para convertirse en proveedor en el sistema.
Referencias	RF 4
Precondiciones	El cliente debe tener permisos para acceder a esta parte del sistema.
Poscondiciones	Información de la solicitud de cuenta de usuario, registrada en la BD.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El cliente accede a la interfaz de Solicitud de cuenta de usuario.	<p>2. El sistema muestra la interfaz que explica las condiciones de uso de la cuenta y los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre. • Apellidos. • Dirección de correo electrónico.

	<ul style="list-style-type: none"> • Nombre de la empresa. • Cargo en la empresa. • URL de la empresa. • Descripción.
3. El cliente presiona el botón guardar.	<p>4. El sistema valida datos.</p> <p>5. El sistema guarda los datos.</p> <p>6. El sistema muestra un mensaje informando al cliente que ha sido registra su solicitud de cuenta.</p> <p>7. El sistema envía un correo al cliente para confirmar la dirección, y finaliza el caso de uso.</p>
Curso alternativo	
Acción 4	4. El sistema muestra un mensaje de error indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Prioridad:	Crítico

Tabla 2.5 Descripción del CU Solicitar cuenta de usuario

2.6.3 Paquetes Gestión de información.

2.6.3.1 Diagrama de caso de uso.

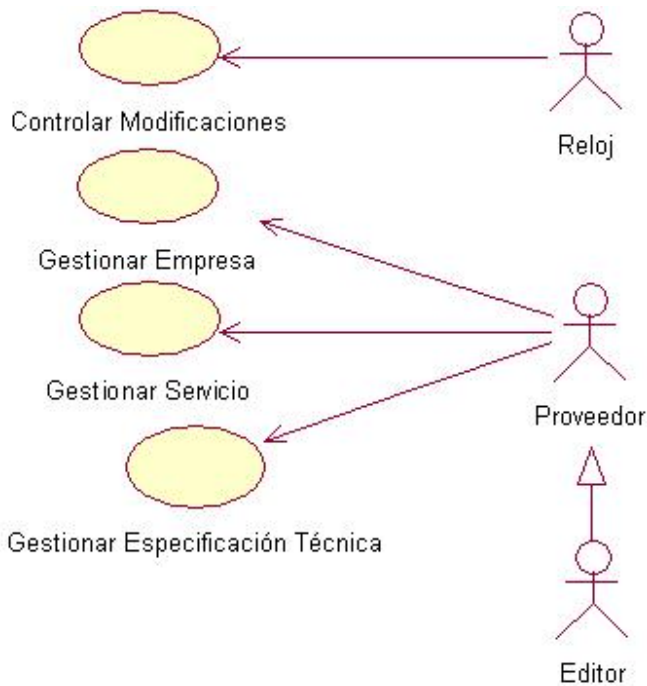


Figura 2.4 Diagrama de CU Paquete Gestión de información.

2.6.3.2 Descripción de los casos de usos.

Nombre del Caso de Uso	Gestionar Empresa.
Actores	Proveedor
Propósito	Registrar, modificar y eliminar una empresa.
Resumen	En este caso de uso el proveedor podrá registrar, modificar y eliminar una empresa.
Referencias	RF 6, 7, 8, 9, 10, 11.
Precondiciones	El proveedor debe tener permisos para acceder a esta parte del sistema
Poscondiciones	<ol style="list-style-type: none"> 1. Información de la empresa registrada en la BD. 2. Información de la empresa modificada en la BD.

	3. Información de la empresa eliminada en la BD.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El proveedor selecciona una de las opciones de la interfaz Gestionar Empresa.	2. El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> a) Si decide adicionar una empresa, ver sección “Registrar Empresa”. b) Si decide ver la empresa, ir a la sección “Mostrar Empresa”.
Sección “Registrar Empresa”.	
	3. El sistema muestra una interfaz con los siguientes datos: <ul style="list-style-type: none"> • Nombre de la empresa. • Descripción de la empresa. • Categoría. • Identificador de empresa. • URL de la empresa.
4. El proveedor llena los datos y presiona el botón Adicionar Contacto.	5. El sistema muestra en la interfaz los datos de los contactos: <ul style="list-style-type: none"> ○ Nombre del contacto. ○ Correo electrónico. ○ Teléfono. ○ Dirección ○ Dirección alternativa.
6. El proveedor llena los datos y presiona el botón Guardar.	7. El sistema valida los datos. 8. El sistema guarda los datos. 9. El sistema muestra un mensaje informando que la empresa ha sido registrada. 10. El sistema va a la sección “Mostrar Empresa” .

Curso Alternativo	
Acción 4	
4. El proveedor selecciona la Ayuda.	5. El sistema muestra una interfaz de ayuda.
Acción 7	7. El sistema muestra un mensaje de error indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Sección "Mostrar Empresa".	
	3. El sistema busca la empresa y la muestra en una interfaz.
4. El proveedor selecciona la opción a realizar.	5. <ul style="list-style-type: none"> a) Si selecciona Adicionar servicio, ir al caso de uso "Gestionar servicio" sección "Registrar servicio". b) Si selecciona un Servicio, ir al caso de uso "Gestionar servicio" sección "Mostrar servicio". c) Si selecciona Adicionar Especificación técnica, ir al caso de uso "Gestionar Especificación técnica" sección "Registrar Especificación técnica". d) Si selecciona una Especificación técnica ir al caso de uso Gestionar Especificación técnica" sección "Mostrar Especificación técnica". e) Si selecciona el botón modificar empresa, ir a la sección "Modificar empresa". f) Si selecciona el botón eliminar empresa, ir a la sección "Eliminar empresa".
Sección "Modificar empresa".	
	5. El sistema muestra una interfaz con todos los datos editables de la empresa.

6. El proveedor llena los campos y presiona el botón Guardar.	7. El sistema valida los datos. 8. El sistema guarda la información de la empresa 9. El sistema muestra un mensaje informando que la empresa ha sido modificada y finaliza el caso de uso.
Curso Alternativo	
Acción 6	
6. El proveedor presiona el botón Modificar Contacto.	7. El sistema muestra una interfaz los datos de los contactos.
8. El proveedor llena los campos y presiona el botón Guardar.	9. Continuar en Acción 6.
Acción 6	
6. El proveedor presiona el botón Eliminar Contacto.	7. El sistema pide confirmación para eliminar los contactos.
8. El proveedor presiona el botón Aceptar.	9. El sistema elimina la empresa. 10. El sistema muestra un mensaje informando que el contacto ha sido eliminado y finaliza el caso de uso.
Acción 7	
	7. El sistema muestra un mensaje de error indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Sección "Eliminar empresa".	
	4. El sistema pide confirmación para eliminar la empresa.
5. El proveedor presiona el botón Aceptar.	6. El sistema elimina la empresa y finaliza el caso de uso.
Prioridad:	Crítico

Tabla 2.6 Descripción del CU Gestionar Empresa.

Nombre del Caso de Uso	Gestionar Servicio.
Actores	Proveedor
Propósito	Registrar, modificar y eliminar un servicio y adicionar un Punto de acceso.
Resumen	En este caso de uso el proveedor podrá registrar, modificar y eliminar un servicio así como registrar puntos de acceso.
Referencias	RF 12, 13, 14, 15, 16, 17.
Precondiciones	El proveedor debe tener permisos para acceder a esta parte del sistema
Poscondiciones	3.1 Información del servicio registrado en la BD. 4.1 Información del servicio modificada en la BD. 5.1 Información del servicio eliminada en la BD.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El proveedor selecciona una de las opciones de la interfaz Gestionar Servicio.	2. El sistema ejecuta alguna de las siguientes acciones: a) Si decide adicionar un servicio, ver sección “Registrar Servicio” . b) Si de decide ver Servicio, ver sección “Mostrar Servicio” .
Sección “Registrar Servicio”.	
	3. El sistema muestra una interfaz que contiene los datos del servicio a registrar, que son: <ul style="list-style-type: none"> • Lista de empresa. • Nombre del servicio. • Descripción del servicio. • Categoría.

<p>4. El proveedor llena los datos del servicio.</p> <p>5. EL proveedor presiona el botón Adicionar Referencia.</p>	<p>6. El sistema valida los datos.</p> <p>7. El sistema muestra en la interfaz los datos de la Referencia a registrar:</p> <ul style="list-style-type: none"> • Tipo de Punto de Acceso. • URL del Punto de Acceso. • PC_remota. • Categoría. • Descripción.
<p>8. El proveedor presiona el botón Guardar.</p>	<p>9. El sistema valida los datos.</p> <p>10. El sistema guarda la información del servicio y la referencia.</p> <p>11. El sistema muestra un mensaje informando que servicio ha sido registrado.</p> <p>12. El sistema va a la sección “Mostrar Servicio”.</p>
<p>Curso Alternativo</p>	
<p>Acción 4</p>	
<p>4. El proveedor selecciona la Ayuda.</p>	<p>5. El sistema muestra una interfaz de ayuda.</p>
<p>Acción 6, 9.</p>	<p>El sistema muestra un mensaje de error indicando que los campos obligatorios tienen caracteres extraños o están vacíos.</p>
<p>Sección “Mostrar Servicio”.</p>	
	<p>3. El sistema busca y muestra una interfaz que contiene el servicios seleccionado.</p>
<p>4. El proveedor selecciona la opción a realizar.</p>	<p>5.</p> <ul style="list-style-type: none"> a) Si selecciona el botón modificar empresa, ir a la sección “Modificar Servicio”. b) Si selecciona el botón eliminar empresa, ir a la sección “Eliminar Servicio”.

	c) Si selecciona una Referencia, ir a la sección “Mostrar Referencia” .
Sección “Modificar Servicio”.	
	6. El sistema muestra una interfaz con todos los datos editables del servicio.
7. El proveedor llena los campos. 8. El usuario presiona el botón Guardar.	9. El sistema valida los datos. 10. El sistema guarda la información del servicio. 11. El sistema muestra un mensaje informando que el servicio ha sido modificado y finaliza el caso de uso.
Sección “Eliminar Servicio”.	
	5. El sistema pide confirmación para eliminar el servicio.
6. El proveedor presiona el botón Aceptar.	7. El sistema elimina el servicio. 8. El sistema muestra un mensaje informando que el servicio ha sido eliminado y finaliza el caso de uso
Sección “Mostrar Referencia”	
	6. El sistema busca y muestra una interfaz que la referencia seleccionada.
7. El proveedor selecciona la opción a realizar.	8. a) Si selecciona el botón modificar empresa, ir a la sección “Modificar Referencia” . b) Si selecciona el botón eliminar empresa, ir a la sección “Eliminar Referencia” .
Sección “Modificar Referencia”	
	9. El sistema muestra una interfaz con todos los datos editables de la referencia.
10. El proveedor llena los campos y presiona el botón Guardar.	11. El sistema valida los datos. 12. El sistema guarda la información.

	13. El sistema muestra un mensaje informando que la referencia ha sido modificado y finaliza así el caso de uso.
Sección “Eliminar Referencia”	
	9. El sistema pide confirmación para eliminar la referencia.
10. El proveedor presiona el botón Aceptar.	11. El sistema elimina la referencia. 12. El sistema muestra un mensaje informando que la referencia ha sido eliminado y finaliza así el caso de uso.
Prioridad:	Crítico

Tabla 2.7 Descripción del CU Gestionar Servicio.

Nombre del Caso de Uso	Gestionar Especificación técnica
Actores	Proveedor
Propósito	Registrar, modificar y eliminar las Especificaciones técnicas.
Resumen	En este caso de uso el proveedor podrá registrar, modificar y eliminar una Especificación técnica.
Referencias	RF 18, 19, 20.
Precondiciones	El proveedor debe tener permisos para acceder a esta parte del sistema
Poscondiciones	1. Información de la especificación técnica registrada en la BD. 2. Información de la especificación técnica modificada en la BD. 3. Información de la especificación técnica eliminada en la BD.
Curso Normal de los Eventos	

Acciones del Actor	Respuesta del Sistema
1. El proveedor selecciona una de las opciones de la interfaz Gestionar Especificación técnica.	2. El sistema ejecuta alguna de las siguientes acciones: a) Si decide adicionar un Especificación técnica, ver sección “Registrar Especificación técnica” . b) Si decide mostrar una Especificación técnica, ver sección “Mostrar Especificación técnica” .
Sección “Registrar Especificación técnica”.	
	3. El sistema muestra una interfaz con los siguientes datos: <ul style="list-style-type: none"> • Nombre de la Especificación técnica. • Descripción. • Categoría. • Identificador de la Especificación técnica. • URL de la Especificación técnica. • Tipo de URL de la Especificación técnica. • Parámetros de instancias. • Lista de servicios.
4. El proveedor llena los datos de la especificación y presiona el botón Guardar.	5. El sistema valida los datos. 6. El sistema guarda la información. 7. El sistema muestra un mensaje informando que la Especificación técnica ha sido registrada y finaliza el caso de uso.
Curso Alternativo	
Acción 4	
6. El proveedor selecciona la Ayuda.	7. El sistema muestra una interfaz de ayuda.
Acción 5	5. El sistema muestra un mensaje de error

	indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Sección “Mostrar Especificación técnica”.	
	3. El sistema busca y muestra una interfaz que la referencia seleccionada.
4. El proveedor selecciona la opción a realizar.	5. <ul style="list-style-type: none"> a) Si el botón modificar Especificación técnica, ver sección “Modificar Especificación técnica”. b) Si el botón eliminar Especificación técnica, ver sección “Eliminar Especificación técnica”.
Sección “Modificar Especificación técnica”.	
	3. El sistema muestra una interfaz con todos los datos editables de la Especificación técnica.
4. El proveedor llena los campos y presiona el botón Guardar	5. El sistema valida los datos. 6. El sistema guarda la información del Especificación técnica. 7. El sistema muestra un mensaje informando que la Especificación técnica ha sido modificada y finaliza el caso de uso.
Curso Alternativo	
Acción 7	7. El sistema muestra un mensaje de error indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Sección “Eliminar Especificación técnica”.	
	3. El sistema pide confirmación para eliminar la Especificación técnica.
4. El proveedor presiona el botón	5. El sistema elimina el Especificación técnica.

Aceptar.	6. El sistema muestra un mensaje informando que la Especificación técnica ha sido eliminada y finaliza el caso de uso
Prioridad:	Crítico

Tabla 2.8 Descripción del CU Gestionar Especificación Técnica.

2.6.4 Paquetes Gestionar Administración.

2.6.4.1 Diagrama de caso de uso.

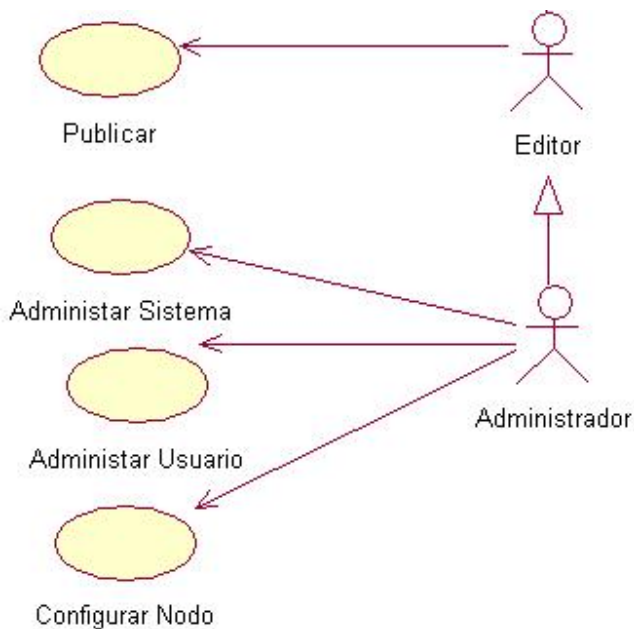


Figura 2.4 Diagrama de CU Paquete Administración.

2.6.4.2 Descripción de los casos de usos.

Nombre del Caso de Uso	Publicar
Actores	Editor
Propósito	Permitir publicar las empresas, servicios y especificaciones técnicas registrados por los proveedores.

Resumen	En este caso de uso el editor se encarga de publicar las empresas, servicios y especificaciones técnicas registrados por los proveedores.
Referencias	RF 22
Precondiciones	El editor debe tener permisos para acceder a esta parte del sistema
Poscondiciones	<ol style="list-style-type: none"> 1. Información de la empresa, publicada en la BD. 2. Información del servicio, publicada en la BD. 3. Información de la especificación técnica, publicada en la BD. 4. Información de la empresa denegada para los usuarios. 5. Información del servicio denegado para los usuarios 6. Información de la especificación técnica denegada para los usuarios.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El editor selecciona una de las opciones de la interfaz Publicar.	<ol style="list-style-type: none"> 2. El sistema ejecuta alguna de las siguientes acciones: <ol style="list-style-type: none"> a) Si decide publicar una empresa, ir a la sección "Publicar Empresa". b) Si decide publicar un servicio, ir a la sección "Publicar Servicio". c) Si decide publicar una Especificación técnica, ir a la sección "Publicar Especificación técnica". d) Si decide denegar una empresa, ir a la sección "Deshabilitar Empresa". e) Si decide denegar un servicio, ir a la sección "Deshabilitar Servicio".

	f) Si decide denegar una Especificación técnica, ir a la sección “Deshabilitar Especificación técnica”.
Sección “Publicar Empresa”.	
	3. El sistema muestra todas las empresas que han sido registradas pero que no están revisadas.
4. El editor selecciona la empresa a publicar.	5. El sistema muestra los datos de la empresa.
6. El editor comprueba los datos de la empresa. 7. El editor presiona el botón OK.	8. El sistema habilita los datos. 9. El sistema muestra un mensaje informando que la empresa ha sido habilitada y finaliza así el caso de uso.
Curso Alternativo.	
Acción 7.	
7. El editor presiona el botón Cancelar.	8. El sistema envía un correo al proveedor de esa empresa explicándole que los datos de la empresa registrada están incorrectos y que tiene un tiempo limite para arreglar los datos.
Sección “Publicar Especificación técnica”.	
	3. El sistema muestra todas las Especificación técnica que han sido registradas pero que no están revisadas.
4. El editor selecciona la empresa a publicar.	5. El sistema muestra los datos de la Especificación técnica.
6. El editor comprueba los datos de la empresa. 7. El editor presiona el botón OK.	8. El sistema habilita los datos. 9. El sistema muestra un mensaje informando que la Especificación técnica ha sido habilitada y finaliza así el caso de uso.
Curso alternativo	

Acción 7.	
7. El editor presiona el botón Cancelar.	8. El sistema envía un correo al proveedor de ese servicio explicándole que los datos de la Especificación técnica registrada están incorrectos y que tiene un tiempo limite para arreglar los datos.
Sección “Publicar servicio”.	
	3. El sistema muestra todos los servicios que han sido registradas pero que no están revisadas.
4. El editor selecciona el servicio a publicar.	5. El sistema muestra los datos del servicio.
6. El editor comprueba los datos del servicio. 7. El editor presiona el botón OK.	8. El sistema registra habilita los datos. 9. El sistema muestra un mensaje informando que el servicio ha sido habilitada y finaliza así el caso de uso.
Curso alternativo	
Acción 7.	
7. El editor presiona el botón Cancelar.	8. El sistema envía un correo al proveedor del servicio explicándole que los datos del servicio registrado están incorrectos y que tiene un tiempo limite para arreglar los datos.
Sección “Deshabilitar Servicio”	
	3. Ir al caso de uso “ Buscar ” sección “ Buscar servicio ”.
4. El editor selecciona el Servicio.	5. El sistema muestra los datos.
6. El editor comprueba los datos y presiona el botón Denegar.	7. El sistema pide confirmación.

<p>8. El editor presiona el botón Aceptar.</p>	<p>9. El sistema deniega el servicio. 10. El sistema muestra un mensaje informando que el servicio ha sido denegado. 11. El sistema envía un correo al proveedor de esa Especificación técnica explicándole que la información publicada presenta problemas y que tiene un tiempo limite para arreglar los datos, terminado el caso de uso.</p>
<p>Sección “Deshabilitar Empresa”</p>	
	<p>3. Ir al caso de uso “Buscar” sección “Buscar empresa”.</p>
<p>4. El editor selecciona una Empresa.</p>	<p>5. El sistema muestra los datos.</p>
<p>6. El editor comprueba los datos y presiona el botón Denegar.</p>	<p>7. El sistema pide confirmación.</p>
<p>8. El editor presiona el botón Aceptar.</p>	<p>9. El sistema deniega la empresa. 10. El sistema muestra un mensaje informando que la empresa ha sido denegada. 11. El sistema envía un correo al proveedor de esa Especificación técnica explicándole que la información publicada presenta problemas y que tiene un tiempo limite para arreglar los datos, terminado el caso de uso.</p>
<p>Sección “Deshabilitar Especificación técnica”.</p>	
	<p>3. Ir al caso de uso “Buscar” sección “Buscar Especificación técnica”.</p>
<p>4. El editor selecciona una Especificación técnica.</p>	<p>5. El sistema muestra los datos.</p>
<p>6. El editor comprueba que los datos y presiona el botón Denegar.</p>	<p>7. El sistema pide confirmación.</p>

<p>8. El editor presiona el botón Aceptar.</p>	<p>9. El sistema deniega la Especificación técnica. 10. El sistema muestra un mensaje informando que a Especificación técnica ha sido denegada. 11. El sistema envía un correo al proveedor de esa Especificación técnica explicándole que la información publicada presenta problemas y que tiene un tiempo limite para arreglar los datos, terminado el caso de uso.</p>
<p>Prioridad:</p>	<p>Crítico</p>

Tabla 2.10 Descripción del CU Publicar.

<p>Nombre del Caso de Uso</p>	<p>Administrar sistema</p>
<p>Actores</p>	<p>Administrador</p>
<p>Propósito</p>	<p>Crear categorías e indicadores para empresas y especificaciones técnicas, y denegar las empresas, servicios y especificaciones técnicas.</p>
<p>Resumen</p>	<p>En este caso de uso el administrador crea las categorías e indicadores de las empresas y servicios, y denegar las empresas, servicios y especificaciones técnicas registrados en el sistema.</p>
<p>Referencias</p>	<p>RF 2</p>
<p>Precondiciones</p>	<p>El administrador debe tener permisos para acceder a esta parte del sistema</p>
<p>Poscondiciones</p>	<ol style="list-style-type: none"> 1. Información del identificador de la empresa registrada en la BD. 2. Información del identificador de la especificación técnica registrada en la BD. 3. Información de la categoría de la empresa registrada en la BD. 4. Información de la categoría de referencia

	<p>registrada en la BD.</p> <p>5. Información de la categoría de la especificación técnica registrada en la BD.</p>
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
<p>1. El administrador selecciona una de las opciones de la interfaz Administrar sistema.</p>	<p>2. El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Si decide crear un identificador de empresa, ir a la sección “Registrar identificador de Empresa”. b) Si decide crear un identificador de especificaciones técnicas, ir a la sección “Registrar identificador de especificaciones técnicas”. c) Si decide crear una categoría de empresa, ir a la sección “Registrar categoría de Empresa”. d) Si decide crear una categoría de referencia, ir a la sección “Registrar categoría de Referencia”. e) Si decide crear una categoría de especificaciones técnicas, ir a la sección “Registrar categoría de Especificaciones técnicas”.
Sección “Registrar identificador de Empresa”	
	<p>3. El sistema muestra una interfaz con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre del identificador. • Descripción.

4. El administrador llena los datos y presiona el botón Crear.	5. El sistema válida los datos. 6. El sistema guarda los datos. 7. El sistema muestra un mensaje informando que identificador ha sido registrado y finaliza así el caso de uso.
Curso Alternativo	
Acción 5	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Acción 6	El sistema muestra un mensaje de error informando que ya existe ese identificador.
Sección “Registrar identificador de Especificación técnica”	
	3. El sistema muestra una interfaz con los siguientes datos: <ul style="list-style-type: none"> • Nombre del identificador. • Descripción.
4. El administrador llena los datos y presiona el botón Crear.	5. El sistema válida los datos. 6. El sistema guarda los datos. 7. El sistema muestra un mensaje informando que identificador ha sido registrado y finaliza así el caso de uso.
Curso Alternativo	
Acción 5	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Acción 6	El sistema muestra un mensaje de error informando que ya existe ese identificador.
Sección “Registrar categoría de Empresa”	

	<p>3. El sistema muestra una interfaz con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre de la categoría. • Descripción.
<p>4. El administrador llena los datos presiona el botón Crear.</p>	<p>5. El sistema válida los datos.</p> <p>6. El sistema guarda los datos.</p> <p>7. El sistema muestra un mensaje informando que la categoría ha sido registrado y finaliza así el caso de uso.</p>
Curso Alternativo	
Acción 5	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Acción 6	El sistema muestra un mensaje de error informando que ya existe esa categoría.
Sección “Registrar categoría de Referencia”	
	<p>3. El sistema muestra una interfaz con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre de la categoría. • Descripción.
<p>4. El administrador llena los datos presiona el botón Crear.</p>	<p>5. El sistema válida los datos.</p> <p>6. El sistema guarda los datos.</p> <p>7. El sistema muestra un mensaje informando que la categoría ha sido registrado y finaliza así el caso de uso.</p>
Curso Alternativo	
Acción 5	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.

Acción 6	El sistema muestra un mensaje de error informando que ya existe esa categoría.
Sección “Registrar categoría de Especificación técnica”	
	3. El sistema muestra una interfaz con los siguientes datos: <ul style="list-style-type: none"> • Nombre de la categoría. • Descripción.
4. El administrador llena los datos presiona el botón Crear.	5. El sistema valida los datos. 6. El sistema guarda los datos. 7. El sistema muestra un mensaje informando que la categoría ha sido registrado y finaliza así el caso de uso.
Curso Alternativo	
Acción 5	5. El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Acción 6	6. El sistema muestra un mensaje de error informando que ya existe esa categoría.
Prioridad:	Crítico

Tabla 2.11 Descripción del CU Administrar Sistema.

Nombre del Caso de Uso	Administrar usuario
Actores	Administrador
Propósito	Revisar, registrar, modificar, buscar y ver el historial de los usuarios del sistema.
Resumen	En este caso de uso el administrador revisa, registra, modifica, buscar y ve el historial de los usuarios del sistema.
Referencias	RF 3

Precondiciones	El usuario debe tener permisos para acceder a esta parte del sistema
Poscondiciones	<ol style="list-style-type: none">1. Información del usuario, registrada en la BD2. Información del usuario revisado, registrada en la BD.3. Información del usuario bloqueado, registrada en la BD.4. Información del usuario desbloqueado, registrada en la BD5. Información del usuario eliminado en la BD.6. Información del usuario buscado, mostrada al administrador.7. Información del historial de usuario, mostrada al administrador.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema

<p>1. El administrador selecciona una de las opciones de la interfaz Administrar usuario.</p>	<p>2. El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none">a) Si decide crear un nuevo usuario, ir a la sección “Registrar usuario”.b) Si decide revisar solicitudes de usuario, ir a la sección “Revisar solicitud”.c) Si decide bloquear un usuario ir a la sección “Bloquear usuario”.d) Si decide desbloquear un usuario ir a la sección “Desbloquear usuario”.e) Si decide eliminar un usuario, ir a la sección “Eliminar usuario”.f) Si decide buscar un usuario, ir a la sección “Buscar usuario”.g) Si decide ver el historial del usuario, ir a la sección “Ver historial de usuario”.
<p>Sección “Registrar Usuario”.</p>	
	<p>3. El sistema muestra un interfaz con los siguientes datos:</p> <ul style="list-style-type: none">• Nombre.• Apellidos.• Usuario del sistema.• Contraseña.• Confirmación de contraseña.• Dirección de correo electrónico.• Rol de usuario.<ul style="list-style-type: none">○ Proveedor.○ Editor.○ Administrador.

4. El administrador llena los datos y presiona el botón Guardar.	5. El sistema valida los datos. 6. El sistema guarda los datos. 7. El sistema muestra un mensaje informando el usuario ha sido registrado. 8. El sistema envía un correo electrónico con su usuario y contraseña y finaliza así el caso de uso.
Curso Alternativo	
Acción 5	5. El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Acción 6	6. El sistema muestra un mensaje de error informando que ya existe ese usuario.
Sección “Revisar Usuario”.	
	3. El sistema muestra en la interfaz con la lista de todas las solicitudes de usuarios.
4. El administrador selecciona una solicitud.	5. El sistema muestra la solicitud.
6. El administrador comprueba los datos de la solicitud. 7. El administrador presiona el botón Aceptar.	8. Ir a la sección “Registrar Usuario” . 9. El sistema elimina la solicitud de usuario del registro.
Curso Alternativo.	
Acción 7.	
7. El administrador presiona el botón Cancelar.	8. El sistema envía un correo al usuario explicándole el que su solicitud ha sido denegada. 9. El sistema elimina la solicitud del registro.
Sección “Bloquear Usuario”.	

	3. Ir a la sección “Buscar Usuario” .
4. El administrador selecciona el usuario.	5. El sistema muestra en interfaz los datos del usuario.
6. El administrador presiona el botón Bloquear.	7. El sistema pide confirmación
8. El administrador presiona el botón Aceptar.	12. El sistema bloquea el usuario. 13. El sistema muestra un mensaje informando que el usuario ha sido bloqueado. 9. El sistema envía un correo al usuario del sistema informando que su cuenta ha sido bloqueada, y así termina el caso de uso.
Sección “Desbloquear Usuario”.	
	10. Ir a la sección “Buscar Usuario” .
11. El administrador selecciona el usuario.	12. El sistema muestra en interfaz los datos del usuario.
13. El administrador presiona el botón Desbloquear.	14. El sistema pide confirmación
15. El administrador presiona el botón Aceptar.	14. El sistema desbloquea el usuario. 15. El sistema muestra un mensaje informando que el usuario ha sido bloqueado. 16. El sistema envía un correo al usuario del sistema informando que su cuenta ha sido bloqueada, y así termina el caso de uso.
Sección “Buscar usuario”.	
	3. El sistema muestra una interfaz con los siguientes parámetros de búsqueda: <ul style="list-style-type: none"> • Nombre. • Usuario. • Rol de usuario.

	<ul style="list-style-type: none"> ○ Proveedor. ○ Editor. ○ Administrador. • Ordenar por nombre: <ul style="list-style-type: none"> ○ Ascendente. ○ Descendente.
4. El usuario llena los datos.	5. El sistema valida los datos. 6. El sistema busca los usuarios. 7. El sistema muestra el resultado de la búsqueda que contiene: <ul style="list-style-type: none"> • Nombre. • Usuario. • Correo Electrónico. • Rol. y finaliza el caso de uso.
Sección “Eliminar Usuario”.	
	3. Ir a la sección “Buscar Usuario” .
4. El administrador selecciona el usuario.	5. El sistema muestra una lista con los usuarios del sistema.
6. El usuario selecciona el usuario y presiona el botón Eliminar.	7. El sistema pide confirmación para eliminar.
8. El usuario presiona el botón Aceptar.	9. El sistema elimina el usuario y finaliza el caso de uso.
Sección “Ver historial”	
	3. Ir a la sección “Buscar Usuario” .

<p>4. El administrador selecciona el usuario y el botón Mostrar.</p>	<p>5. El sistema muestra en la interfaz los siguientes datos:</p> <ul style="list-style-type: none"> • Usuario del sistema • Fecha de creada la cuenta de usuario. • Fecha de entrada al sistema. • Hora de entrada al sistema. • Fecha de salida al sistema. • Hora de salida al sistema. <p>y finaliza así el caso de uso.</p>
<p>Prioridad:</p>	<p>Crítico</p>

Tabla 2.12 Descripción del CU Administrar Usuario.

Nombre del Caso de Uso	Configurar un nodo
Actores	Administrador
Propósito	Realizar la configuración de un nodo.
Resumen	En este caso de uso el administrador introduce los datos necesarios para configurar el nodo.
Referencias	RF 21.
Precondiciones	El administrador debe tener permisos para acceder a esta parte del sistema
Poscondiciones	Información del nodo registrada en la BD.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
<p>1. El administrador solicita configurar el nodo.</p>	<p>2. El sistema muestra una interfaz con los siguientes datos:</p> <ul style="list-style-type: none"> • Nodo <ul style="list-style-type: none"> ○ Nombre del Nodo operador. ○ Descripción. ○ Generador de llave.

	<ul style="list-style-type: none"> ○ Nombre del host. ○ Puerto del host. ○ Dirección del nodo. ○ Nombre de versión. • Cuenta <ul style="list-style-type: none"> ○ Nombre de usuario. ○ Contraseña. ○ Confirmación de contraseña.
3. El usuario llena los datos	4. El sistema verifica la validez de los datos. 1. El sistema guarda los datos y finaliza así el caso de uso.
Curso alternativo	
Acción 4.	El sistema muestra un mensaje indicando que los campos obligatorios tienen caracteres extraños o están vacíos.
Prioridad:	Crítico

Tabla 2.13 Descripción del CU Configurar Nodo.

Nombre del Caso de Uso	Controlar modificaciones
Actores	Reloj
Propósito	Eliminar todas las empresas que están deshabilitadas.
Resumen	En este caso de uso el reloj informa la fecha al sistema, este revisa todas las empresas, servicios y especificaciones técnicas con los estados: revisado y deshabilitado, y que su fecha limite coincide con la del reloj, y las elimina de la BD.
Referencias	RF 23

Precondiciones	<p>Información de las empresas estén en estado revisado y deshabilitado.</p> <p>Información de los servicios estén en estado revisado y deshabilitado.</p> <p>Información de las especificaciones técnicas estén en estado revisado y deshabilitado.</p>
Poscondiciones	<p>Información de la empresa, eliminada de la BD.</p> <p>Información del servicio, eliminada de la BD.</p> <p>Información de la especificación técnica, eliminada de la BD.</p>
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
El reloj informa al sistema de la fecha.	<p>El sistema busca todas empresas, servicios y especificaciones técnicas con los estados: revisado y deshabilitado, y que su fecha limite coincide con la del reloj.</p> <p>El sistema elimina cada una de las empresas, servicios y especificaciones técnicas encontradas.</p> <p>El sistema envía a cada proveedor un mensaje explicándole que sus datos fueron eliminados.</p>
Prioridad:	Crítico

Tabla 2.14 Descripción del CU Controlar Modificaciones.

2.7 Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución, obteniéndose a partir del análisis del modelo del dominio, un listado con las funciones que debe tener el sistema, que se representaron mediante un Diagrama de Casos de Uso, y finalmente se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Gracias a esto ahora se puede empezar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funciones que han sido consideradas necesarias en este capítulo.

CAPÍTULO
3
Análisis y Diseño

3.1 Introducción

Tras la definición y descripción, en el anterior capítulo, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrollará.

Este capítulo tiene el objetivo de plantear la concepción general del diseño del sistema propuesto. Así, se presentan los diagramas de clases Web que detallan la interacción de las distintas páginas y se estructura la información que se desea persista a través del diseño de la base de datos. Son también descritos los estándares de diseño seguidos.

3.2 Modelo de Análisis

El objetivo de realizar el modelo de análisis es conseguir una comprensión mas precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que sirva de ayuda para estructurar el sistema entero.

En el análisis se analizan los requisitos con mayor profundidad y se utiliza el lenguaje del desarrollador para describir los resultados. Esto da la ventaja de razonar más sobre los aspectos internos del sistema, y por tanto resolver elementos relativos a la interferencia de casos de uso.

Esboza como llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura, sirve como una primera aproximación al diseño.

El modelo de análisis esta compuesto por clase de análisis, que representan una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Posee un conjunto de características tales como: se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, define atributos y participa en relaciones, entre otras. [29]

Cada clase posee uno de los estereotipos siguientes:

- **Clases de interfaz:** se utilizan para modelar la interacción entre el sistema y sus actores, la cual implica recibir (y presentar) información y peticiones desde (y hacia) los usuarios.
- **Clases de entidad:** se utilizan para modelar información que posee una vida larga o que es a menudo persistente, modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto, o un suceso del mundo real.
- **Clases de control:** Coordinan la realización de uno o unos pocos CU, coordinando las actividades de los objetos que implementan la funcionalidad del CU.

Cumpliendo con lo anterior analizado y por la importancia en su información se definirán las entidades del modelo de análisis.

- E_Empresa: entidad que contiene los datos de la empresa como el nombre de la empresa, descripción, URL de la empresa.
- E_Contacto: entidad que contiene los datos de los contactos de una empresa como nombre del contacto, dirección.
- E_Servicio: entidad que contiene los datos de los servicios que ofrece una empresa como son nombre del servicio, descripción.
- E_Referencia: entidad que contiene los datos que referencian a un servicio como es la URL del punto de acceso.
- E_Especif_Técnica: entidad que contiene los datos de las especificaciones técnicas de un servicio como el nombre de la especificación, descripción.
- E_Infor_Especif_Técnica: entidad que contiene los datos de información de una especificación técnica como la URL de la especificación, parámetros de instancia.

3.2.1 Paquete Usuario.

3.2.1.1 Caso de Uso: Autenticar.

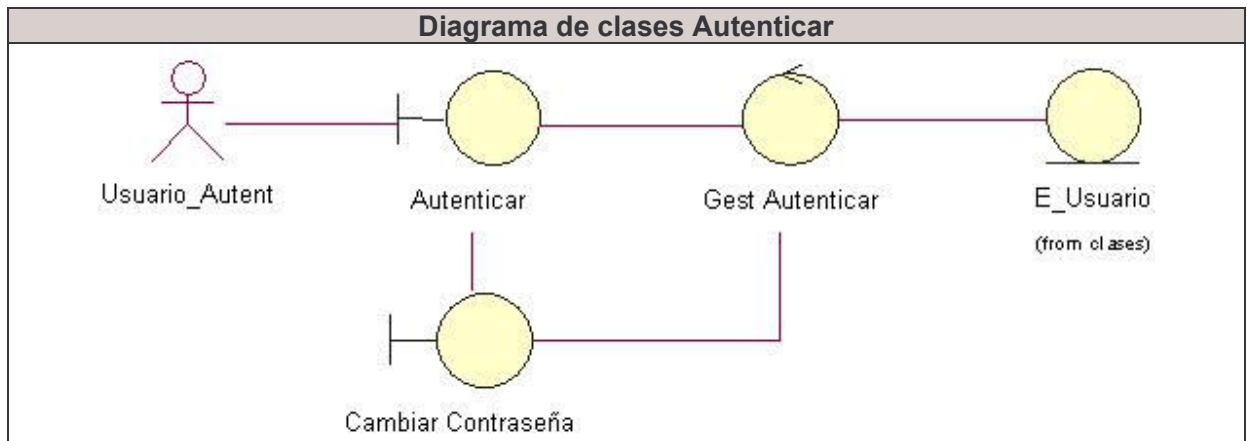


Figura 3.1 Diagrama de clases de análisis del CU Autenticar

3.2.2 Paquete Petición

3.2.2.1 Caso de Uso: Buscar.

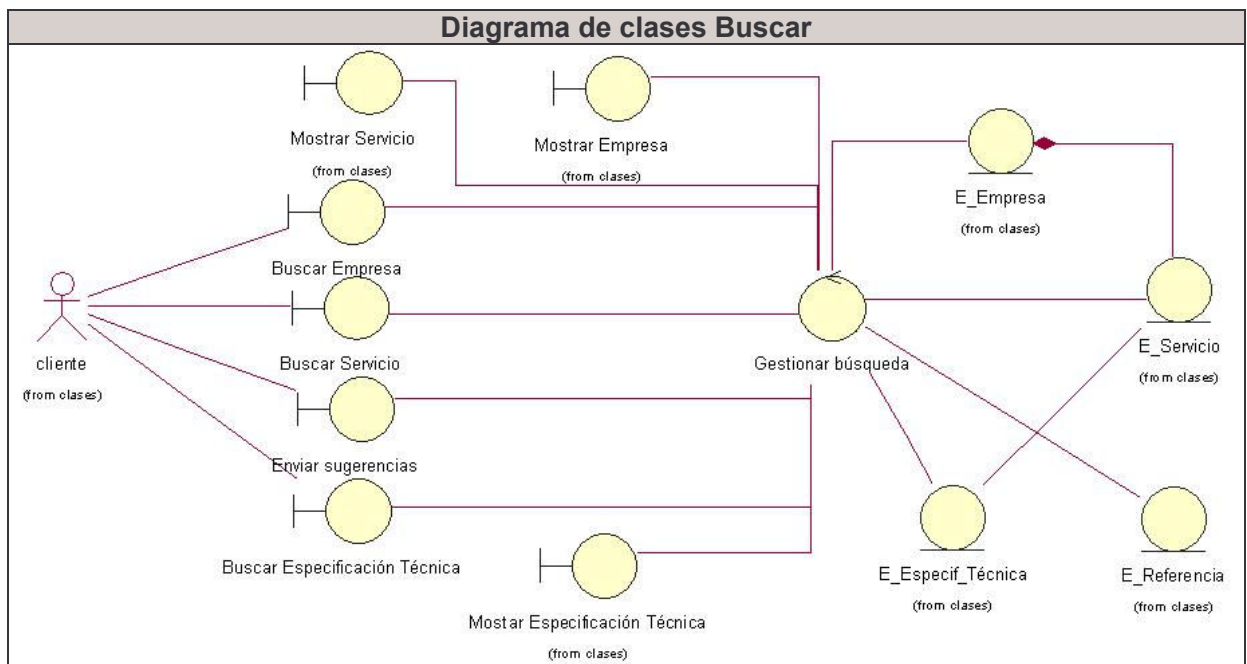


Figura 3.2 Diagrama de clases de análisis del CU Buscar.

3.2.3 Paquete de Gestionar Información.

3.2.3.1 Caso de Uso: Gestionar Empresa.

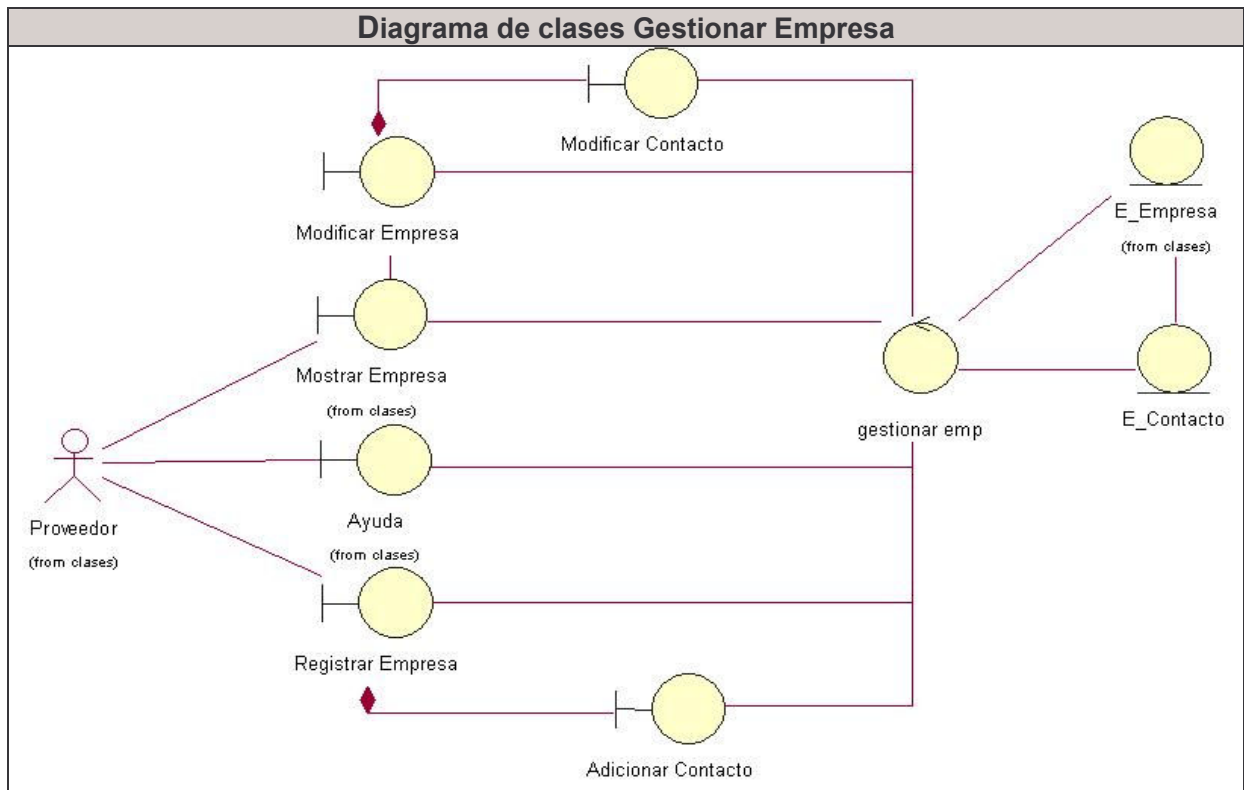


Figura 3.3 Diagrama de clases de análisis del CU Gestionar Empresa.

3.2.4 Paquete Administración

3.2.4.1 Caso de Uso: Administrar Usuario.

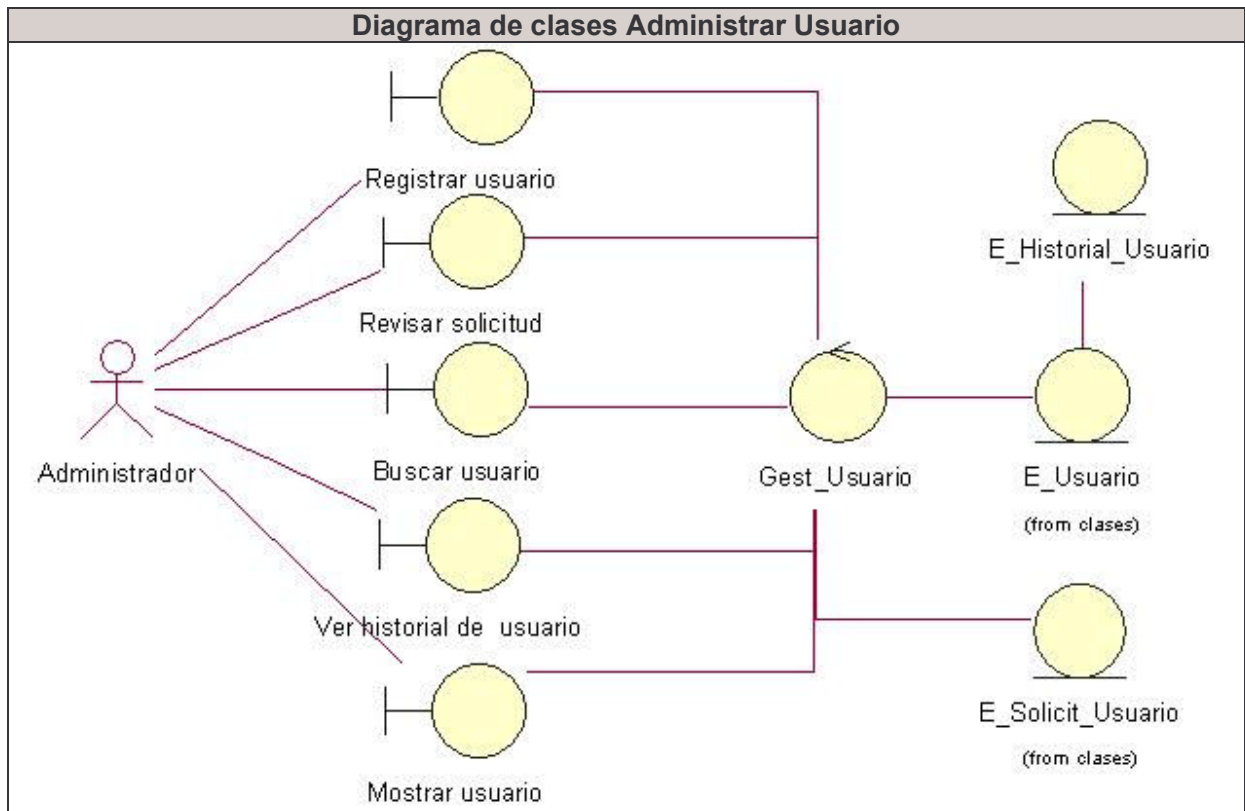


Figura 3.4 Diagrama de clases de análisis del CU Administrar Servicio.

Los restantes diagramas de clases desarrollados se encuentran en el Anexo.

3.3 Modelo de Diseño.

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además el modelo de diseño sirve de abstracción de la implementación del sistema, y es de ese modo, utilizada como una entrada fundamental de las actividades de implementación. [29]

3.3.1 Diagramas de Clases de Diseño.

El diagrama de clases para las Aplicaciones Web, difiere un poco del resto de las aplicaciones que se construyen, puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente; estos son los artefactos que se necesitan modelar para que el desarrollador los implemente luego y obtener así el producto final.

3.3.1.1 Paquete Usuario.

3.3.1.1.1 Caso de Uso: Autenticar.

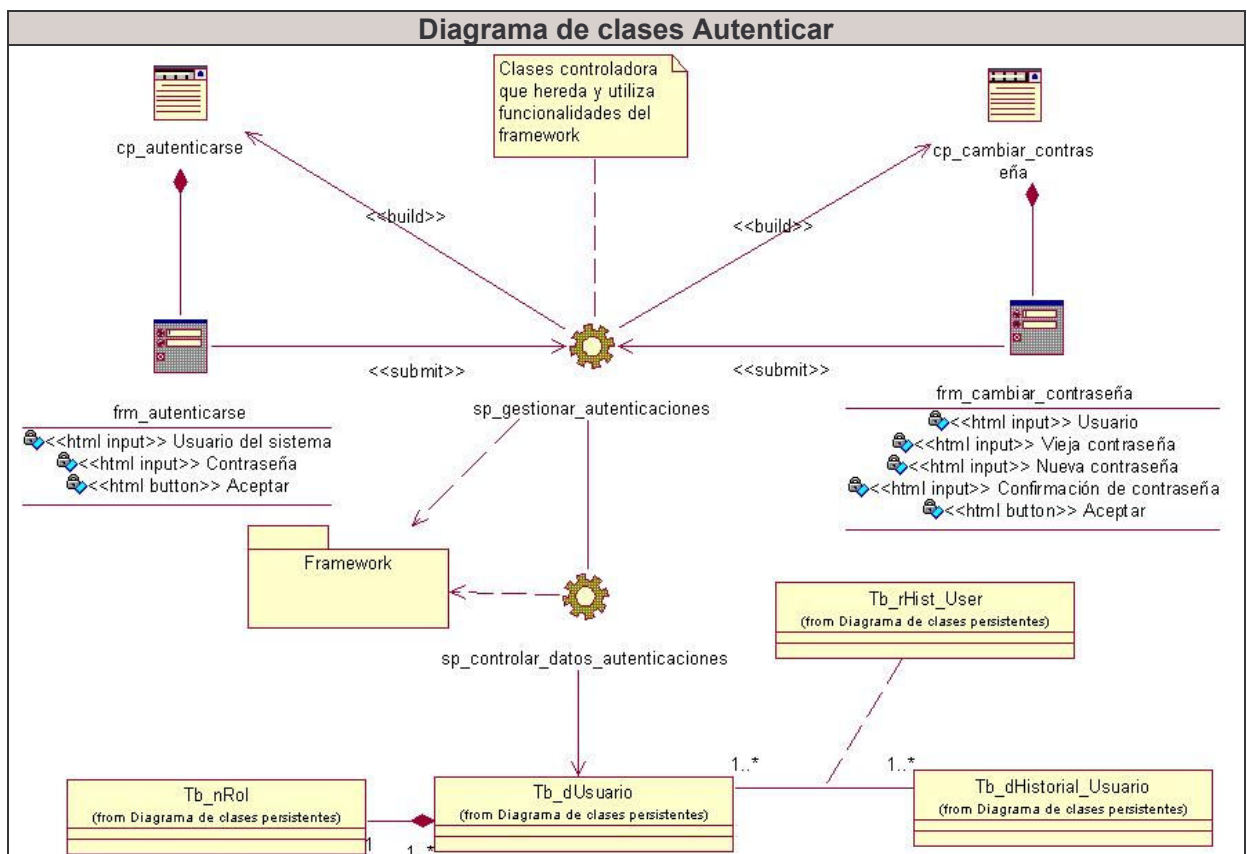


Figura 3.5 Diagrama de clases de diseño del CU Autenticar

3.3.1.2 Paquete Petición

3.3.1.2.1 Caso de Uso: Buscar.

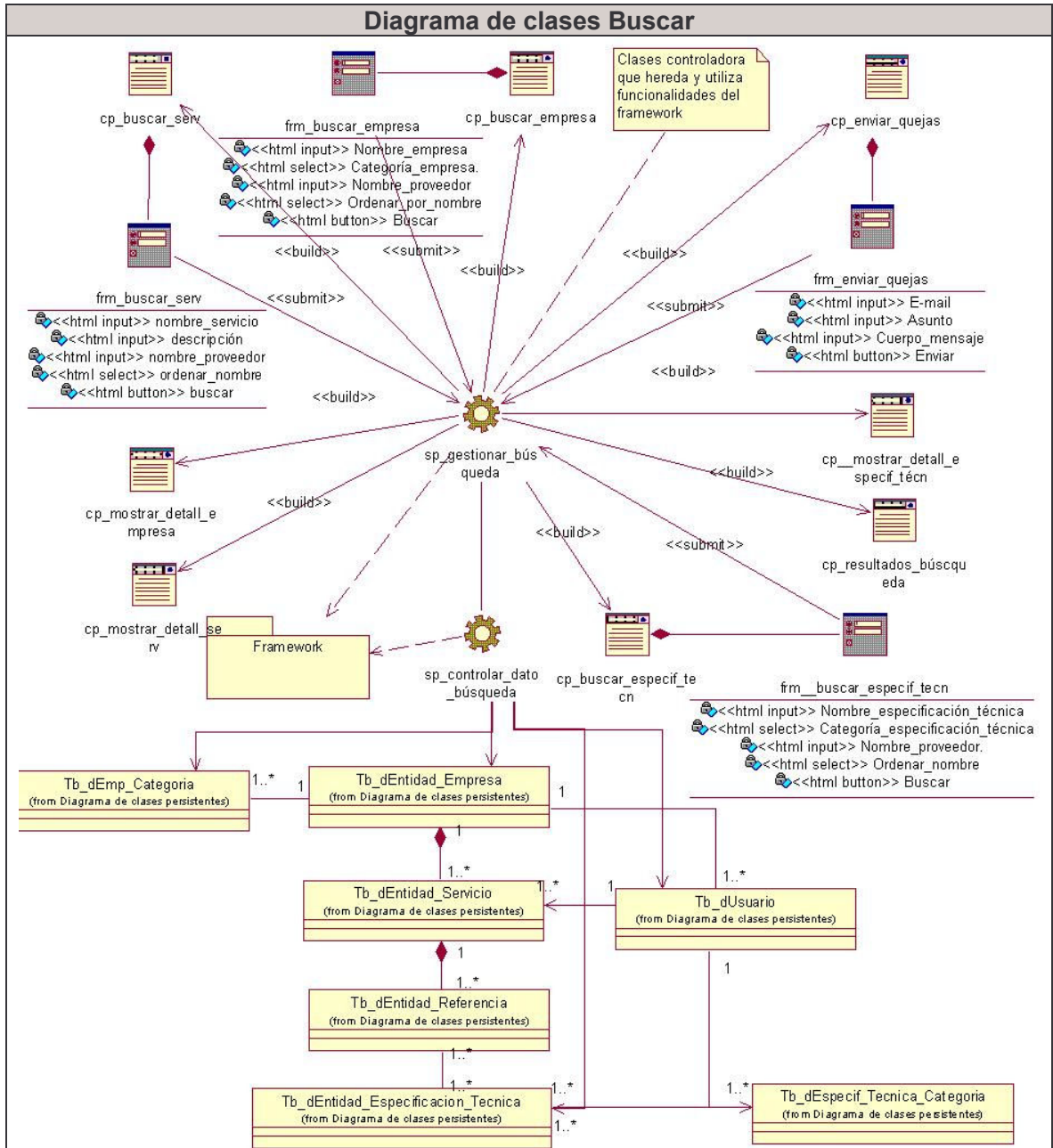


Figura 3.6 Diagrama de clases de diseño del CU Buscar

3.3.1.3 Paquete de Gestionar Información.

3.3.1.3.1 Caso de Uso: Gestionar Empresa.

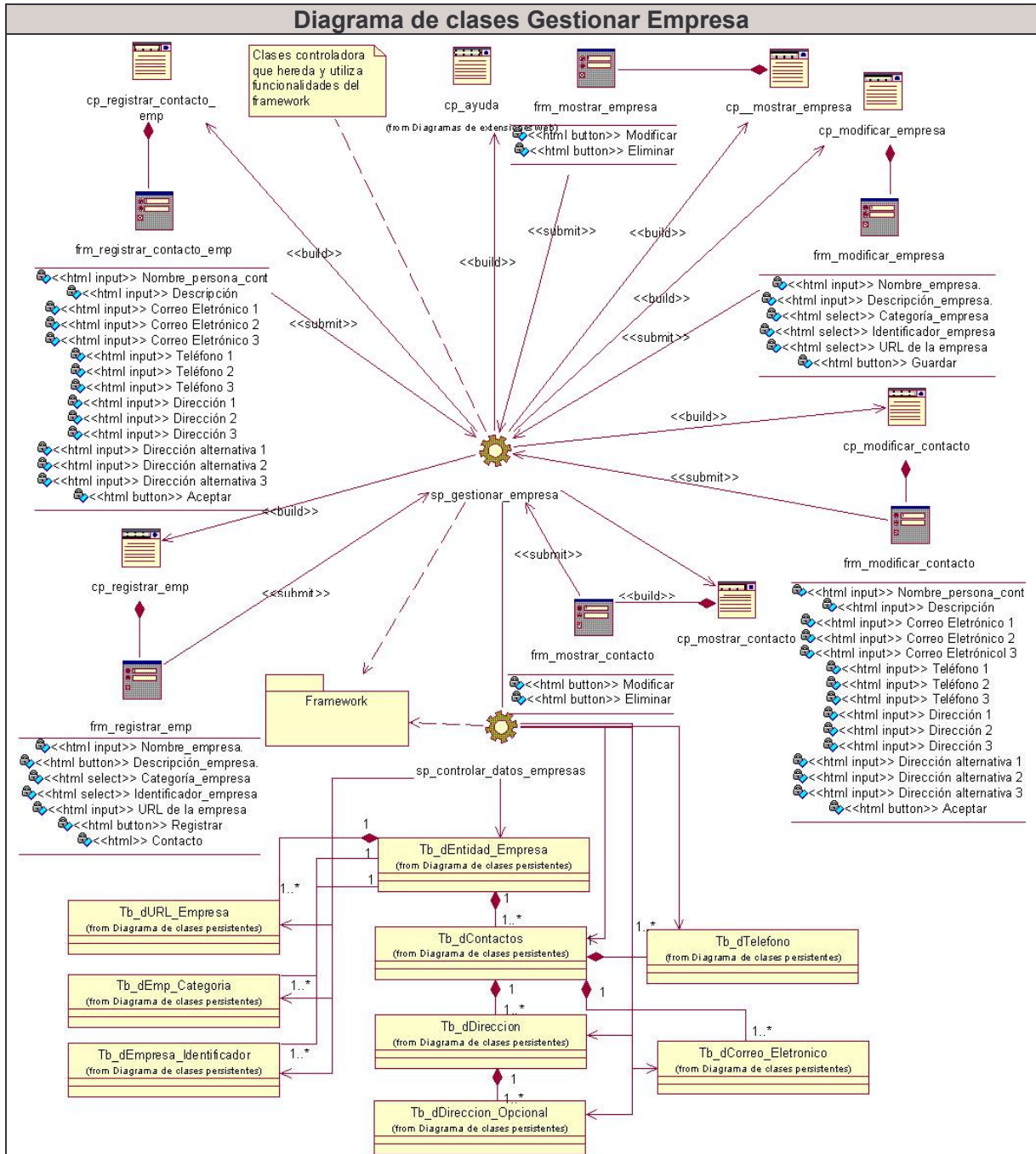


Figura 3.7 Diagrama de clases de diseño del CU Gestionar Empresa

3.3.1.4 Paquete Administración

3.3.1.4.1 Caso de Uso: Administrar Usuario.

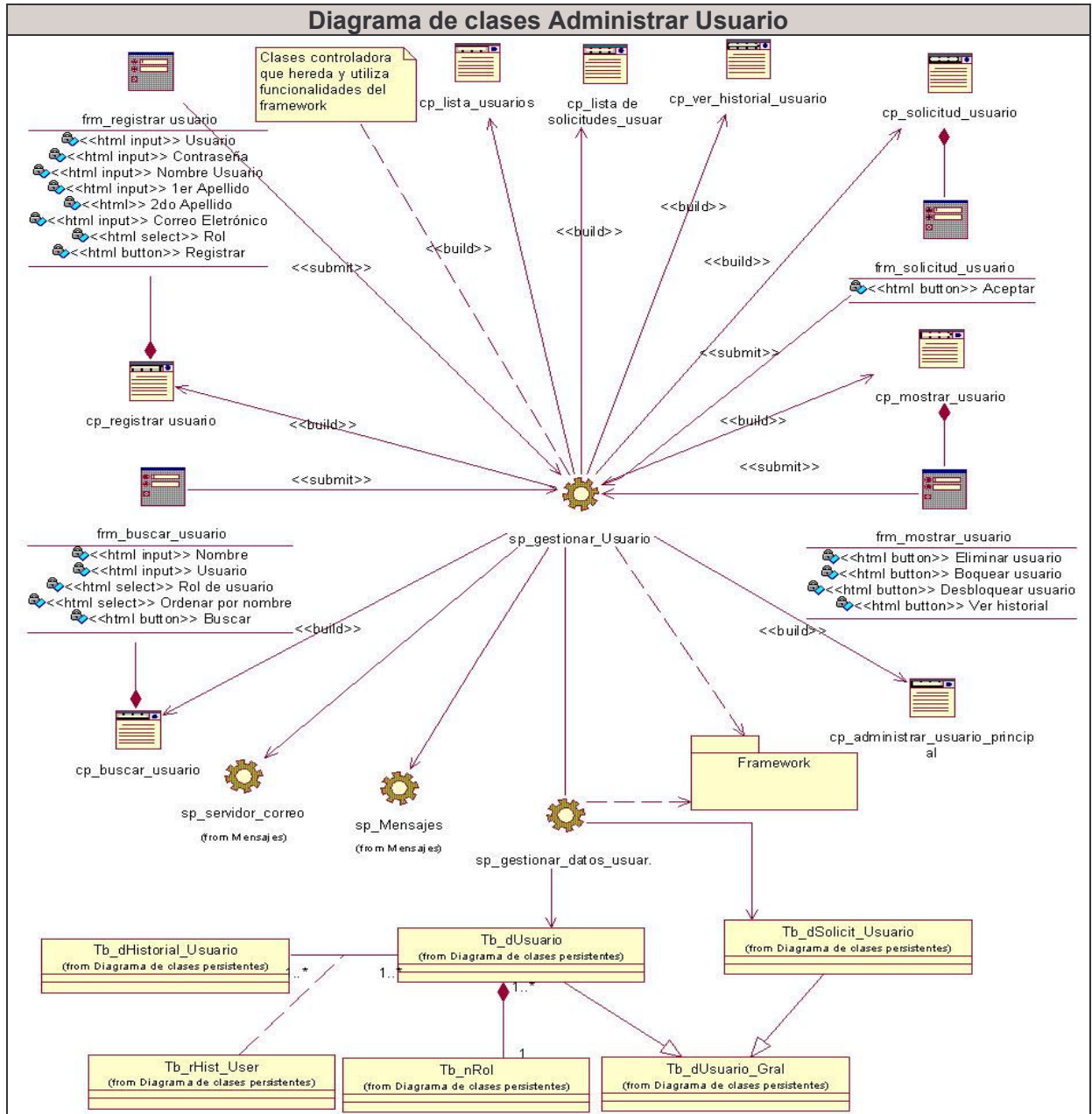


Figura 3.8 Diagrama de clases de diseño del CU Administrar Usuario

Los restantes diagramas de clases desarrollados se encuentran en el Anexo 6.

3.3.2 Diagramas de interacción.

Los diagramas de interacción describen la forma en que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema, mostrando el modo en que los objetos interactúan a través de mensajes. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia, ambos son representaciones alternas de interacciones. Los Diagramas de Secuencia muestran interacciones entre objetos basadas en el tiempo y los Diagramas de Colaboración muestran como los objetos se asocian unos con otros.

3.4 Principios de Diseño.

El diseño de la interfaz de una aplicación, el formato de los reportes, la concepción de la ayuda y el tratamiento de excepciones tiene gran influencia en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión.

3.4.1 Interfaz de usuario

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema, se puede definir como: “el conjunto de trabajos y pasos que seguirá el usuario, durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que realizará, así como las respuestas que el sistema dará”.

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

Una de las premisas fundamentales de la aplicación es la ventaja que proporcionan las interfaces Web sobre las interfaces de comando. Ya que las interfaces Web:

- Proporcionan un ambiente amigable.
- Conducen a un aprendizaje más natural.
- Establecen un “sentimiento” (sobre todo en la uniformidad del ambiente) al usuario que enriquece su experiencia en el uso de la aplicación.

Además de estos principios, se tuvieron en cuenta las siguientes características:

- Utilizar una misma tipografía, forma y estilo en todas las páginas.
- La facilidad del usuario de poder navegar desde cualquier punto a otro dentro de la aplicación.
- Se tuvo presente siempre el ancho de banda y por ello se utilizaron formato de imágenes de compresión favorables.
- La simplicidad y consistencia, favoreciendo la usabilidad de la aplicación.
- Navegación simple en todas las páginas de la aplicación, de forma tal que siempre sea accesible por el usuario.
- Estabilidad y uniformidad del diseño, para así poder ubicar al usuario dentro del mismo y hacerlo sentir parte de él.

Se utilizó una hoja de estilos para guardar la configuración del diseño para todas las páginas, para los botones y las líneas se utilizaron estos estilos, eliminando así el número de imágenes que demoren la presentación de la página.

Los formularios de entradas ocupan el centro superior y las entradas organizadas por importancia. Se incluye una breve explicación del objetivo del formulario, y alguna especificación con respecto a las entradas.

Se realizan múltiples operaciones en cada página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación, para completar una operación.

3.4.2 Tratamiento de errores.

El tratamiento general de la gestión de error recomendado para los nodos de UDDI es detectar errores en una petición antes de procesar la petición. Cualquier error en la petición detectada invalidará la petición entera, y se muestra al usuario un mensaje de error con texto claro que contiene el tipo y número del error.

Algunas de los mensajes son:

- E_MuchasDatos(10030): Significa que fueron introducidos demasiados argumentos.
- E_noSoportado(10050): Significa que el sistema no soporta esa funcionalidad o API.
- E_usuarioNoDisponible: (10110) Significa que la información del usuario de autenticación no respondió en una cantidad de tiempo razonable.

- E_usuarioRequerido: (10120) Significa que falta o es inválida la información del usuario para la autenticación.
- E_usuarioNoAutorizado(10140): Significa que se intentó utilizar la API de publicación para cambiar datos controlados por otra parte.
- E_noConoceUsuario: (10150) Significa que no reconoce o no son válidos los datos del nombre del usuario y la contraseña introducido.
- E_límiteCuentaExcedido: (10160) Significa que excedió la cantidad de datos en una petición para guardar un tipo de datos.
- E_ocupado(10400): Significa que la petición no puede ser procesada en el tiempo actual.
- E_errorFatal(10500): Significa que un error técnico serio ha ocurrido procesando la petición.
- E_valorInválido(20200): Este error tiene múltiples empleos, se aplica en las suscripciones, esto puede ser usado para indicar que un valor que fue introducido en un atributo no pasó la validación.
- E_valorNoPermitido(20210): Significa que un valor no pasó la validación debido a cuestiones contextuales. El valor puede estar válido en algunos contextos, pero no en el contexto usado.
- E_tiempoMáximoPetición(20240): Significa que la petición no podía ser realizada porque un servicio Web necesario, no respondió en una cantidad de tiempo razonable.
- E_mensajeMuyLargo(30110): Significa que el mensaje es demasiado grande.
- E_datosHistorialNoDisponible(40010): Significa que los datos del historial solicitados no están disponibles durante el período de tiempo solicitado.
- E_tiempoInválido(40030): Significa que el período de tiempo, la fecha/tiempo, o el par de fecha/tiempo son inválidos.
- E_usuarioYaExiste(40070): Significa que un o más de los elementos de una empresa o elementos de Especificación Técnica que identifican las entidades del nodo no están relacionadas con el proveedor actual.
- E_llaveNoDisponible(40100): Significa que la llave propuesta está en una partición que ya ha sido asignada a algún otro editor.
- E_ningúnValorDisponible(40200): Significa que el intento de recuperar valores válidos no devolvió ningún valor.

- E_combinaciónInválida(40500): Significa que los parámetros de búsqueda que se han especificado están en conflicto.
- E_resultadoMuyLargo(40300): Significa que el nodo considera que el resultado de una búsqueda es demasiado grande, y solicita para obtener los resultados subconjuntos de estos.

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en este se registra y muestra. Los errores se tratan en una página especial que incluye el fichero de configuración general, y está preparada para recoger el número del error y presentar la pantalla con el error que le corresponde a ese código.

3.5 Diseño de la base de datos.

La base de datos es el sistema utilizado para el almacenamiento de datos y acceso controlado a los datos almacenados. En este epígrafe se muestra el diseño de la base de datos del sistema propuesto a través del diagrama de clases persistentes y el esquema de la base de datos generados a partir de este, el modelo de datos.

3.5.1 Diagrama de clases persistentes.

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente, la necesidad de guardar su estado esta dado por al almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información. A continuación se muestra el diagrama de clases persistentes.

nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes. El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema.

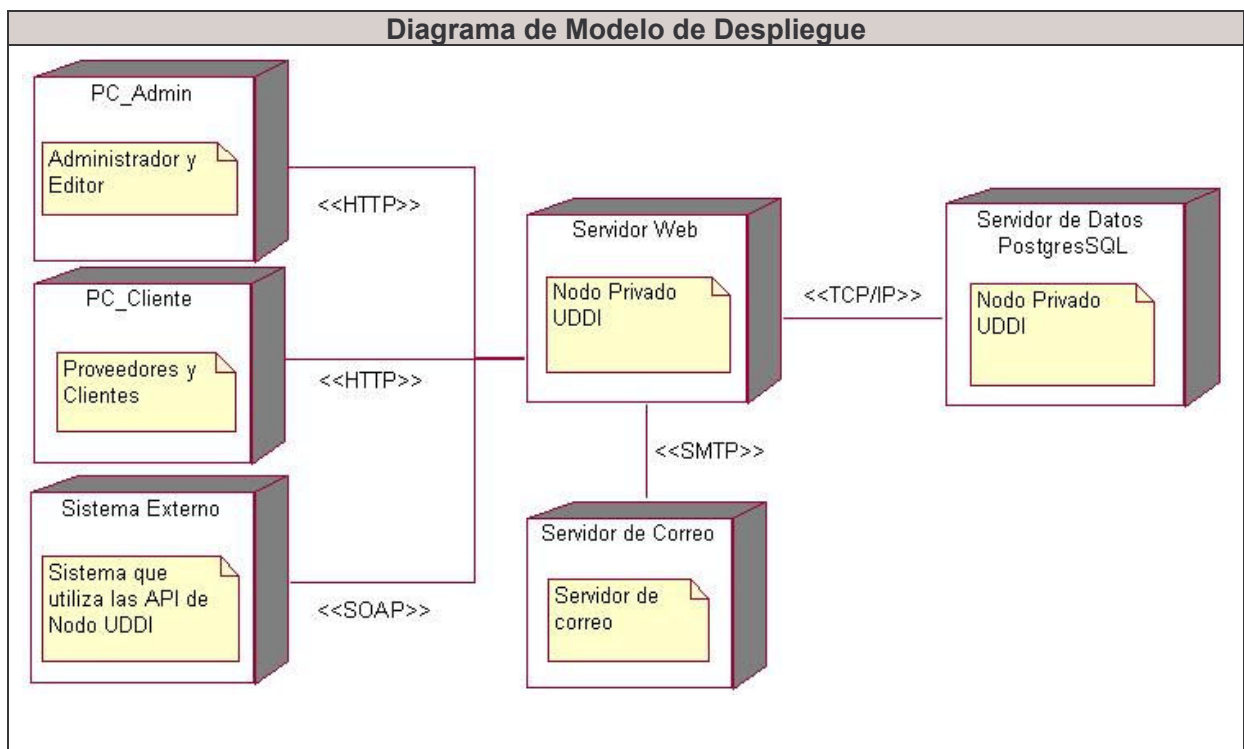


Figura 3.11 Diagrama de Modelo de Despliegue.

3.7 Conclusiones.

En el presente capítulo se mostraron los resultados de la etapa de diseño del sistema. Se desarrollaron los diagramas de interacción a través de los diagramas de secuencia del diseño, para obtener una mayor comprensión del funcionamiento del sistema.

Con los resultados obtenidos en los diagramas de interacción y a modo de evolución del Modelo conceptual, se obtuvo el diagrama de clases donde se representaron las clases de forma que quedó una representación de las clases, sus asociaciones, atributos y responsabilidades. Se describieron, además, los principios de diseño seguidos, específicamente, los temas de estándares de la interfaz, concepción del tratamiento de errores, sistema de ayuda y principios de codificación.

Todos estos elementos obtenidos brindan una idea mucho más clara de las clases, subsistemas, algoritmos, etc. para lograr una mejor codificación.

CAPÍTULO **4**
Estudio de Factibilidad

4.1 Introducción.

Para la realización de un proyecto es de suma importancia el análisis del costo y los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto.

En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios.

4.2 Planificación basada en casos de uso.

Paso 1. Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0

Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	3	9
Total			3	9

Tabla 4.1 Factor de peso de los actores sin ajustar.

$$UAW = \sum \text{cant actores} * \text{peso}$$

$$UAW = 3 * 3$$

$$UAW = 9$$

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	7	35
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	3	30
Complejo	El caso de uso tiene más de 8 transacciones.	15	1	15
Total				80

Tabla 4.2 Factor de peso de los casos de uso sin ajustar.

$$UUCW = \sum \text{cant CU} * \text{Peso}$$

$$UUCW = 80$$

$$UUCP = UAW + UUCW$$

$$UUCP = 9 + 80$$

$$UUCP = 89$$

Paso 2. Cálculo de los Puntos de casos de uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Comentario	Total
T1	Sistema distribuido	2	3	El sistema es centralizado.	6
T2	Tiempo de respuesta	1	2	La velocidad es relativamente alta.	2
T3	Eficiencia del usuario final	1	1	Abundantes restricciones de eficiencia.	1
T4	Funcionamiento Interno complejo	1	1	La complejidad es normal.	1
T5	El código debe ser reutilizable	1	1	Se requiere que el código sea reutilizable.	1
T6	Facilidad de instalación	0,5	1	Algunos requerimientos de facilidad de instalación.	0,5
T7	Facilidad de uso	0,5	3	Elevada.	1,5
T8	Portabilidad	2	1	Se requiere que el sistema sea portable.	2

T9	Facilidad de cambio	1	1	Se requiere un costo relativamente alto de mantenimiento.	1
T10	Concurrencia	1	3	Hay concurrencia.	3
T11	Incluye objetivos especiales de seguridad	1	1	Seguridad Normal	1
T12	Provee acceso directo a terceras partes	1	1	No todos los usuarios web tienen acceso directo.	1
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	Pocos usuarios internos, el sistema es relativamente fácil de usar.	1
Total					22

Tabla 4.3 Factor de complejidad técnica.

$$TCF=0.6+0.01* \sum (\text{peso}*\text{valor asignado})$$

$$TCF=0.6+0.01*22$$

$$TCF=0.6+0.22$$

$$TCF=0.82$$

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Factor	Descripción	Peso	Valor asignado	Comentario	Total
--------	-------------	------	----------------	------------	-------

E1	Familiaridad con el modelo de proyecto utilizado	1,5	0	El grupo está bastante familiarizado con el modelo.	0
E2	Experiencia en la aplicación	0,5	2	La mayoría del grupo ha trabajado mucho tiempo en ésta aplicación.	1
E3	Experiencia en la orientación a objetivos.	1	4	La mayoría del grupo programa en Objetos.	4
E4	Capacidad del analista líder.	0,5	4	Tiene bastante Experiencia.	2
E5	Motivación.	1	5	El grupo está altamente motivado.	5
E6	Estabilidad de requerimientos	2	5	No se esperan muchos cambios.	10
E7	Personal Part-Time	-1	0	Todo el grupo no es full-time.	0
E8	Dificultad del lenguaje de programación	-1	2	Se usará PHP5.	-2
Total					20

Tabla 4.4 Factor de ambiente.

$$EF = 1.4 - 0.03 * \Sigma (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * \Sigma (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * 20$$

$$EF = 1.4 - 0.6$$

$$EF = 0.8$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 89 * 0.82 * 0.8$$

$$UCP = 58,384$$

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombre.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

CF = 20 Horas-Hombre / Punto de Casos de uso.

$$E = UCP * CF$$

$$E = 58,384 * 20$$

$$E = 1167,68 \text{ Horas-Hombre}$$

Paso 4. Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
-----------	--------------	---------------

Análisis	20	583,84
Diseño	20	583,84
Implementación	40	1167,68
Pruebas	10	291,92
Sobrecarga (otras actividades)	10	291,92
Total	100	2919,2

Tabla 4.5 Esfuerzo del proyecto.

Si $E_T = 2919.2$ horas-hombre y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un $E_T = 15.20416$ mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en 17 meses y medio aproximadamente.

-Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 3 hombres

CHM = 3 * Salario Promedio

CHM = 300.00 \$/mes

Costo = CHM * E_T / CH

Costo = 300.00 * 15.20416 / 3

Costo = \$ 1520.416

Tiempo = E_T / CH

Tiempo = 15.20416 / 3

Tiempo = 5.06 ≈ 5 meses

De los resultados obtenidos se interpreta que con 3 hombres trabajando en el proyecto el mismo se desarrolla en 5 meses y su costo total se estima que sea \$1520.416.

4.3 Beneficios tangibles e intangibles.

El Sistema para la informatización del Catálogo de Servicios Web, UDDI no es un producto con fines comerciales, su principal objetivo es resolver los problemas que existen durante el manejo de los servicios web en cada uno de los proyectos que se desarrollan en la universidad.

El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que le permita registrar, actualizar y conocer de una forma más precisa y en el menor tiempo posible la información referente a los servicios web que se van creando en cada proyecto y garantizar con esto la interoperabilidad entre las aplicaciones desarrollada.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- Disminución del tiempo y esfuerzo que se invierte en esta tarea dando a conocer los servicios web que con los que cuenta la universidad.
- Disminución de los gastos pues resulta menos costoso reutilizar los servicios web implementados, que crearlos, garantizando que los sistemas no se atrasen en su entrega.
- Fácil y rápido acceso y publicación de la información actualizada.
- Fácil procesamiento de la información y obtención, dinámica, de los proveedores de la información que se publique en UDDI de cada servicio web.

4.4 Análisis de costos y beneficios.

Desarrollar un producto informático cuesta. Justificar entonces su desarrollo depende de los beneficios que reportarían su implantación y utilización. Los beneficios pueden ser económicos y de orden social, estos últimos son de tanta importancia como los primeros. El sistema que se propone está dirigido a la Universidad de las Ciencias Informáticas, para la gestión de los proyectos que exporta, por tanto su mayor beneficio es de orden económico.

Una vez implantado el sistema éste contribuirá a aumentar la eficiencia en el uso de los servicios web al disminuir el tiempo necesario en que tarde una aplicación en realizarse, garantizando una constante actualización de la información.

La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios; debido a la estructuración en capas de los procesos del negocio que se diseñaron.

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

4.5 Conclusiones.

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

La herramienta propuesta reportará beneficios significativos e importantes para la gestión de los proyectos que se desarrollan en el entorno empresarial, al contribuir a mejorar los sistemas informáticos que se realizan aquí en función de controlar y unificar la información de los servicios web implementados, lo que indica que es factible implementar la herramienta propuesta.

CONCLUSIONES

Hoy en día el uso de los servicios web resulta muy ventajoso al alcanzar la interoperabilidad entre los sistemas conectados en la red, para esto es necesaria una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Sin importar el lenguaje de programación sobre los que estas aplicaciones estén desarrolladas, ni la plataforma sobre la que se ejecuten; siempre que tengan implementados servicios web para comunicarse, lo podrán hacer sin ninguna complicación.

En la universidad se hace necesario establecer un control de los servicios web que se realizan, y quien los realiza, la arquitectura básica de servicios web debe incluir lo necesario para el intercambio de mensajes entre los proveedores, poner en conocimiento la existencia de ese servicio web y que este sea encontrado para utilizarlo.

Una vez estudiados dentro de la pila de protocolos para Servicios Web, el protocolo UDDI, encargado de permitir el descubrimiento y la publicación de los servicios web se llegaron a las siguientes conclusiones:

- El trabajo alcanzó satisfactoriamente los objetivos propuestos: Diseño del Catálogo UDDI (Descripción e Integración Universal del Descubrimiento) en software libre que cumpla con las especificaciones y estándares internacionales.
- Se logró llegar a un entendimiento de las especificaciones internacionales establecidas del catálogo diseñado para obtener un software respaldado en cualquier lugar.
- Se realizó un modelo de datos, donde se almacenará toda la información necesaria que se genera de los procesos relacionados con el control de la información de los servicios web, para de esta forma garantizar la veracidad y centralización de la misma.
- Se logró obtener una solución propuesta acertada, acorde a los requerimientos que soportan el sistema, diseñando casos de uso que satisfacen las necesidades funcionales.
- Se han seguido los principios básicos de diseño descritos para el desarrollo del sistema.
- Se realizó un análisis para garantizar la seguridad y protección de los datos consecuente con el nivel de seguridad requerido.

- Se unificó la información referente al diseño, uso y estándares referentes adaptados a la arquitectura SOA.
- Se demostró que el diseño realizado tiene calidad por lo tanto se puede proceder con confianza a al siguiente flujo.

RECOMENDACIONES

Se recomienda:

- Realizar los flujos de trabajo del RUP que faltan para comprobar la utilidad que tiene el sistema en cualquier centro de trabajo informático.
- Que se ejecuten los requerimientos no funcionales de software con las particularidades necesarias una vez que el sistema sea construido
- Que el sistema sea capaz de generar estadísticas sobre el uso de servicios web, en función del cliente, número de accesos, tipo, etc.
- Que el sistema aproveche la información de logs generada de forma centralizada en la aplicación, para poder controlar los movimientos de cada usuario dentro de este.
- Que se pueda en un futuro adaptar a UDDI a un entorno más específico o centralizado.

REFERENCIA BIBLIOGRÁFICA

1. **Martínez, Leonardo Valcarcel.** *Estudio UDDI*. UCI. La Habana : s.n., 2004. pág. 87, Tesis de diplomado. Tesis investigativa.
2. **Robert Brooke.** Wikipedia.org. [En línea] [Citado el: 3 de 05 de 07.] http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios.
3. **Systinet.** Systinet.com. *Why Does SOA Matter*. [En línea] [Citado el: 3 de 05 de 07.] http://www.systinet.com/soa_explained/why_soa_matters.
4. **Msaffirio.** msaffirio.wordpress.com. [En línea] [Citado el: 16 de 04 de 07.] <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>, ¿Qué son los servicios web?, Mario Saffirio, 5/02/06.
5. **Charles B. Abruzi.** wikipedia.org. *Middleware*. [En línea] [Citado el: 25 de 04 de 07.] <http://es.wikipedia.org/wiki/Middleware>.
6. **Jonathan Mirell.** wikipedia.org. [En línea] [Citado el: 02 de 04 de 07.] <http://es.wikipedia.org/wiki/RPC>.
7. <http://xml.osmosislatina.com>. *http://xml.osmosislatina.com*. [En línea] [Citado el: 20 de 14 de 07.] <http://xml.osmosislatina.com/curso/graficas/uddi>.
8. **González, Charles Benjamín.** Una visión general II. *desarrolloweb.com*. [En línea] [Citado el: 20 de 04 de 07.] <http://www.desarrolloweb.com/articulos/1538.php>.
9. Seely, Scott. <http://www.microsoft.com>. [En línea] Octubre de 2002. [Citado el: 13 de 04 de 2007.] <http://www.microsoft.com/spanish/msdn/articulos/archivo/121202/voices/understw.asp>.
10. **Camero, Leyany Alonso.** *Módulo Admisión del Sistema Automatizado para la Gestión de Información de la Misión Milagro*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2006. pág. 143, Trabajo de Diploma.
11. Mandrake, Revisión Rápida de PHP5 integrado con Zend. [En línea] 09 de 04. [Citado el: 27 de 03 de 06.] <http://www.venezolano.web.ve/archives/230-Revision-rapida-de-PHP5->

12. AJAX. Wikipedia. [En línea] 03 de 04. [Citado el: 20 de 03 de 07.]
<http://es.wikipedia.org/wiki/AJAX>.
13. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas*. 2000. pág. 425. Vol. 1.
14. Rational Unified Process. 2003.
15. **Corporation, Microsoft.** [En línea] [Citado el: 21 de 03 de 07.]
<http://www.microsoft.com>.
16. **Greg Werner.** <http://www.razonypalabra.org.mx>. [En línea] [Citado el: 21 de 04 de 07.]
<http://www.razonypalabra.org.mx/antiores/n49/bienal/Mesa%205/XML.pdf>.

BIBLIOGRAFÍA.

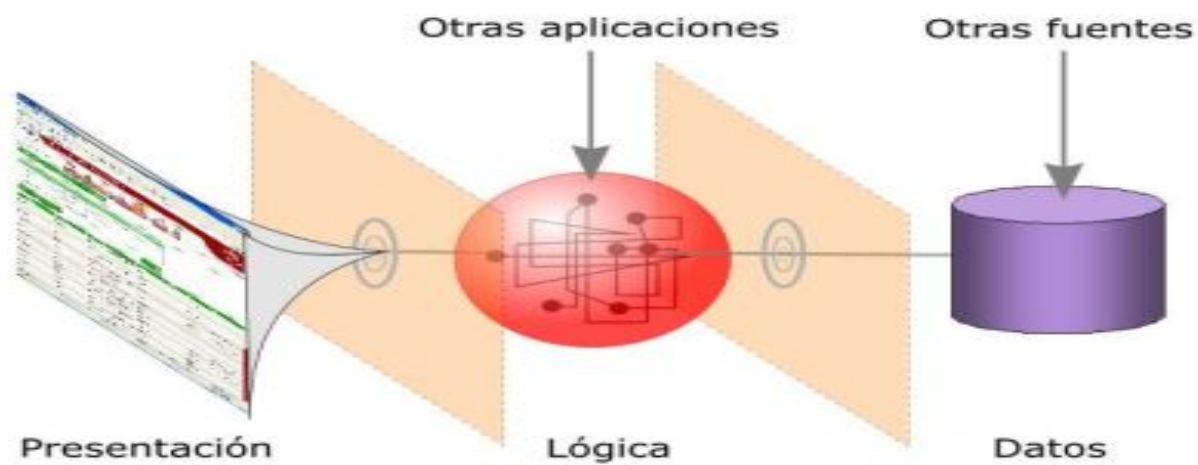
- **AJAX.** *Wikipedia.* [En línea] 03 de 04. [Citado el: 20 de 03 de 07.] [http://es.wikipedia.org/wiki/AJAX.](http://es.wikipedia.org/wiki/AJAX)
- **Cantero, J.** Un vistazo a PHP5 [I]. [En línea] 07 de 04. [Citado el: 11 de 03 de 07.] [http://libertonia.escomposlinux.org/story/2004/7/15/115328/134.](http://libertonia.escomposlinux.org/story/2004/7/15/115328/134)
- **Camero, Leyany Alonso.** *Módulo Admisión del Sistema Automatizado para la Gestión de Información de la Misión Milagro.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2006. pág. 143, Trabajo de Diploma.
- **Corporation, Microsoft.** [En línea] [Citado el: 21 de 03 de 07.] [http://www.microsoft.com.](http://www.microsoft.com)
- **Charles B. Abruzi.** *wikipedia.org. Middleware.* [En línea] [Citado el: 25 de 04 de 07.] [http://es.wikipedia.org/wiki/Middleware.](http://es.wikipedia.org/wiki/Middleware)
- **eusa.ne.** [En línea] [Citado el: 14 de 03 de 07.] [http://www.eusa.net/pages/research/web-services-2.do;jsessionid=771D8082C566966F2CEBF2237EE0213B.](http://www.eusa.net/pages/research/web-services-2.do;jsessionid=771D8082C566966F2CEBF2237EE0213B)
- **González, Charles Benjamín.** Una visión general II. *desarrolloweb.com.* [En línea] [Citado el: 20 de 04 de 07.] [http://www.desarrolloweb.com/articulos/1538.php.](http://www.desarrolloweb.com/articulos/1538.php)
- **Greg Werner.** [http://www.razonypalabra.org.mx.](http://www.razonypalabra.org.mx) [En línea] [Citado el: 21 de 04 de 07.] [http://www.razonypalabra.org.mx/antiores/n49/bienal/Mesa%205/XML.pdf.](http://www.razonypalabra.org.mx/antiores/n49/bienal/Mesa%205/XML.pdf)
- **IBM:UDDI.** IBM Software: Solutions: Web service. [En línea] [Citado el: 15 de 03 de 07.] [http://www-3.ibm.com/services/uddi/.](http://www-3.ibm.com/services/uddi/)
- **IBM Corporation.** [En línea] [Citado el: 25 de 03 de 07.] [http://www.ibm.com.](http://www.ibm.com)
- **Ivar Jacobsn, Grady Booch, James Rumbaugh.** *El Proceso Unificado del Desarrollo del Software.* pág. 434. Vol. 1.
- **Jonathan Mirell.** *wikipedia.org.* [En línea] [Citado el: 02 de 04 de 07.] [http://es.wikipedia.org/wiki/RPC.](http://es.wikipedia.org/wiki/RPC)

- **Larman, Craig.** *UML Y PATRONES Introduccion al analisis y diseño orientado a objetos.* 1999. pág. 499. Vol. 1.
- **Luc Clement, Systinet, y otros.** UDDI Version 3.0.2. *www.oasis.com.* [En línea] [Citado el: 12 de 03 de 2007.] <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
- **msaffirio.** *msaffirio.wordpress.com.* [En línea] [Citado el: 16 de 04 de 07.] <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>, ¿Qué son los servicios web?, Mario Saffirio, 5/02/06.
- Mandrake, Revisión Rápida de PHP5 integrado con Zend. [En línea] 09 de 04. [Citado el: 27 de 03 de 06.] <http://www.venezolano.web.ve/archives/230-Revision-rapida-de-PHP5->.
- **Martínez, Leonardo Valcarcel.** *Estudio UDDI.* UCI. La Habana : s.n., 2004. pág. 87, Tesis de diplomado. Tesis investigativa.
- **Microsoft Corporation.** Microsoft UDDI Business Registry (UBR) node . [En línea] [Citado el: 20 de 03 de 07.] <http://uddi.microsoft.com>.
- **Navarro, Ing. José Angel Franco.** *UML en acción. Modelando Aplicaciones Web.* Ingeniería de Software, CUJAE. La Habana : s.n. pág. 21.
- **Nick A. Monghert.** *wikipedia.org.* [En línea] [Citado el: 20 de 04 de 07.] http://es.wikipedia.org/wiki/Servicio_Web.
- Oasis-UDDI . [En línea] [Citado el: 10 de 03 de 07.] <http://www.uddi.org/specification.html> .
- **Pressman, Roger S.** *Ingenieria del Software. Un enfoque Práctico.* [ed.] Felix Varela. 2005. pág. 614. Vol. 1,2.
- Rational Unified Process. 2003.
- **Robert Brooke.** *Wikipedia.org.* [En línea] [Citado el: 3 de 05 de 07.] http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios.
- **Seely, Scott.** <http://www.microsoft.com>. [En línea] Octubre de 2002. [Citado el: 13 de 04 de 2007.] <http://www.microsoft.com/spanish/msdn/articulos/archivo/121202/voices/understw.asp>.
- **Schmuller, Joseph.** *Aprendiendo UML en 24 horas.* 2000. pág. 425. Vol. 1.

- **Systinet.** Systinet.com. *Why Does SOA Matter.* [En línea] [Citado el: 3 de 05 de 07.] http://www.systinet.com/soa_explained/why_soa_matters.
- UDDI-Central. [En línea] [Citado el: 12 de 04 de 07.] <http://www.uddicentral.com/>.
- Wikipedia. *SOAP.* [En línea] [Citado el: 28 de 03 de 07.] <http://es.wikipedia.org/wiki/SOAP>.
- XML. *Wikipedia.* [En línea] [Citado el: 20 de 04 de 07.] <http://es.wikipedia.org/wiki/XML>.
- <http://xml.osmosislatina.com>. *http://xml.osmosislatina.com.* [En línea] [Citado el: 20 de 14 de 07.] <http://xml.osmosislatina.com/curso/graficas/uddi>.
- <http://xml.osmosislatina.com>. [En línea] [Citado el: 2 de 05 de 2007.] <http://xml.osmosislatina.com/curso/xmlrpc.htm>.
- <http://www.eusa.net/pages/research/web-services-2.do;jsessionid=771D8082C566966F2CEBF2237EE0213B>. [En línea] [Citado el: 13 de 03 de 2007.]

ANEXOS.

Anexo 1: Modelo Cliente Servidor de tres Capas.



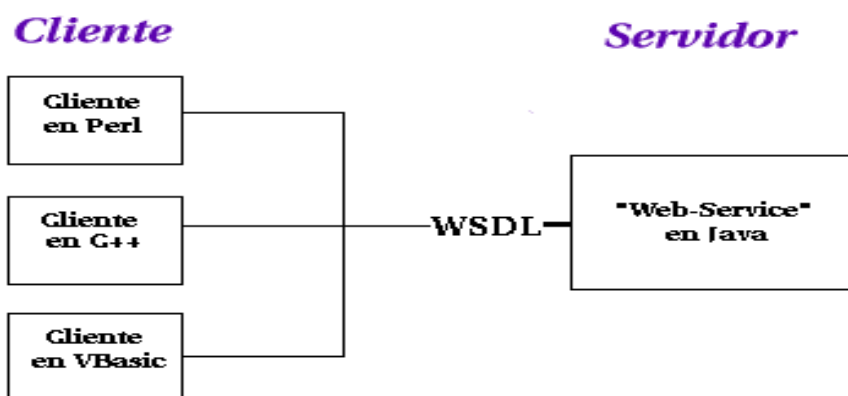
Anexo 2 Ejemplo mensaje SOAP petición.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <catalogo:buscalsbn xmlns:catalogo="http://catalogo.org/cat">
      <catalogo:isbn>
        84-4553-3334-2X
      </catalogo:isbn>
    </catalogo:buscalsbn>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Anexo 3 Ejemplo mensaje SOAP respuesta.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <catalogo:buscalsbnResponse xmlns:catalogo="http://catalogo.org/cat">
      <catalogo:titulo>
        Catalogar materiales especiales
      </catalogo:titulo>
      <catalogo:autor>Marta de Juanes</catalogo:autor>
    </catalogo:buscalsbnResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Anexo 4 Ventajas de WSDL.



Anexo 5 Interoperabilidad Servicios Web en PHP

suma.php

```

<?php
include "nusoap.php";
$namespace = "http://petra.euitio.uniovi.es/~labra/ws/suma.php?wsdl";
$servidor = new soap_server;
$servidor -> configureWSDL ("ServicioSuma", $namespace,
                           "http://petra.euitio.uniovi.es/~labra/ws/suma.php");
$servidor -> wsdl -> schemaTargetNamespace = $namespace;
$servidor -> register ('suma', array ('a' => 'xsd:float', 'b' => 'xsd:float'),
                      array ('return' => 'xsd:float'),
                      'http://petra.euitio.uniovi.es/~labra/ws/suma.php', "", "", "", "");
$servidor -> service ($HTTP_RAW_POST_DATA);

function suma ($a, $b) {
if (!$a || !$b) {
    return new soap_fault ("Client", "", "Se necesitan dos argumentos");
}
}
  
```

```
}  
if ((gettype ($a) != "integer" && gettype ($a) != "double") ||  
    (gettype ($b) != "integer" && gettype ($b) != "double")) {  
    return new soap_fault ("Client", "", "El tipo debe ser entero o real");  
}  
return $a + $b;  
}  
?>
```

Anexo 6 Diagrama de Clases del Modelo de Diseño.

Diagrama de clases Gestionar Servicio

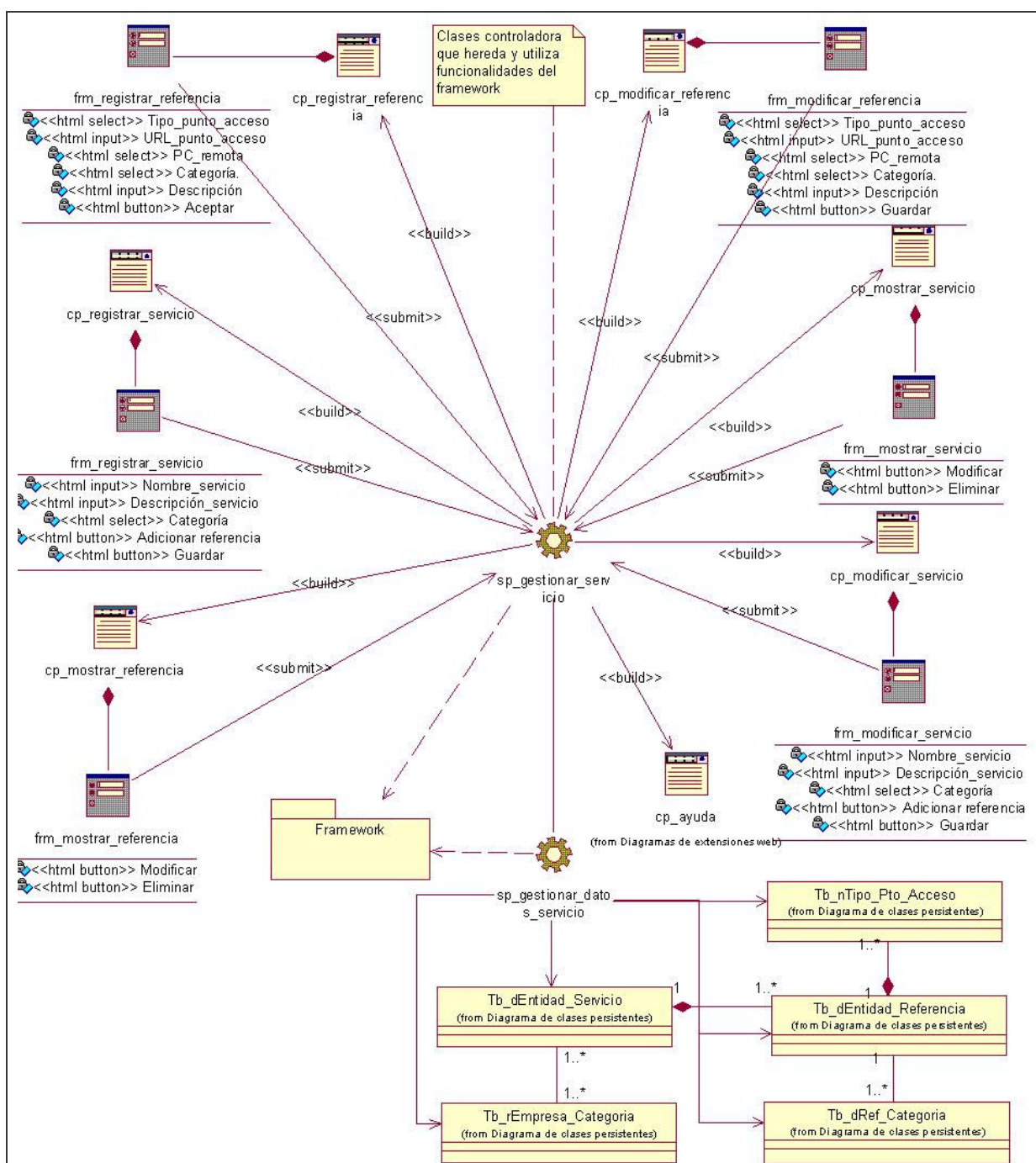


Figura 3.12 Diagrama de clases de diseño del CU Gestionar Servicio

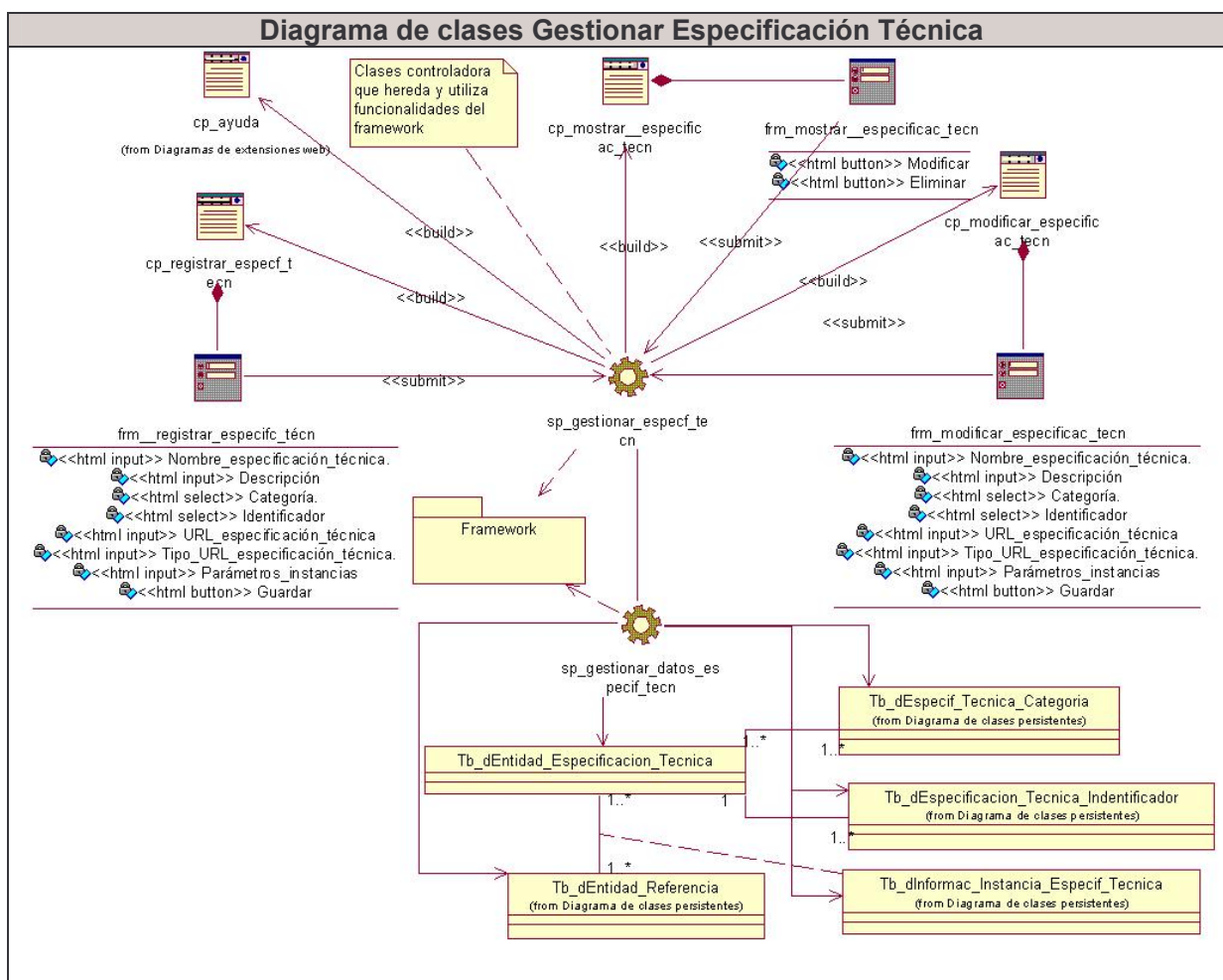


Figura 3.13 Diagrama de clases de diseño del CU Gestionar Especificación Técnica.

Diagrama de clases Gestionar Administrar Sistema

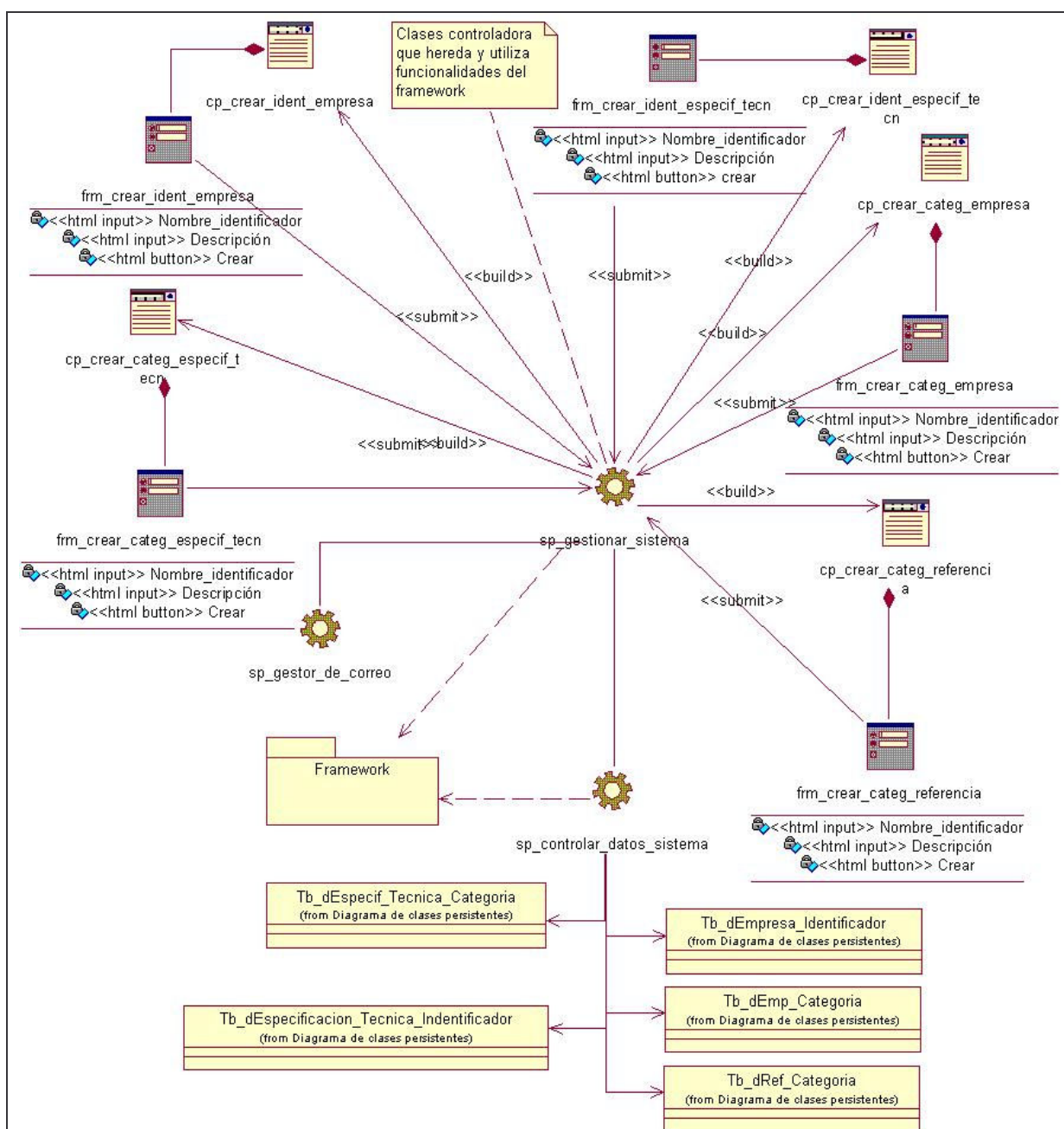


Figura 3.14 Diagrama de clases de diseño del CU Gestionar Administrar Servicio.

Diagrama de clases Gestionar Especificación Técnica

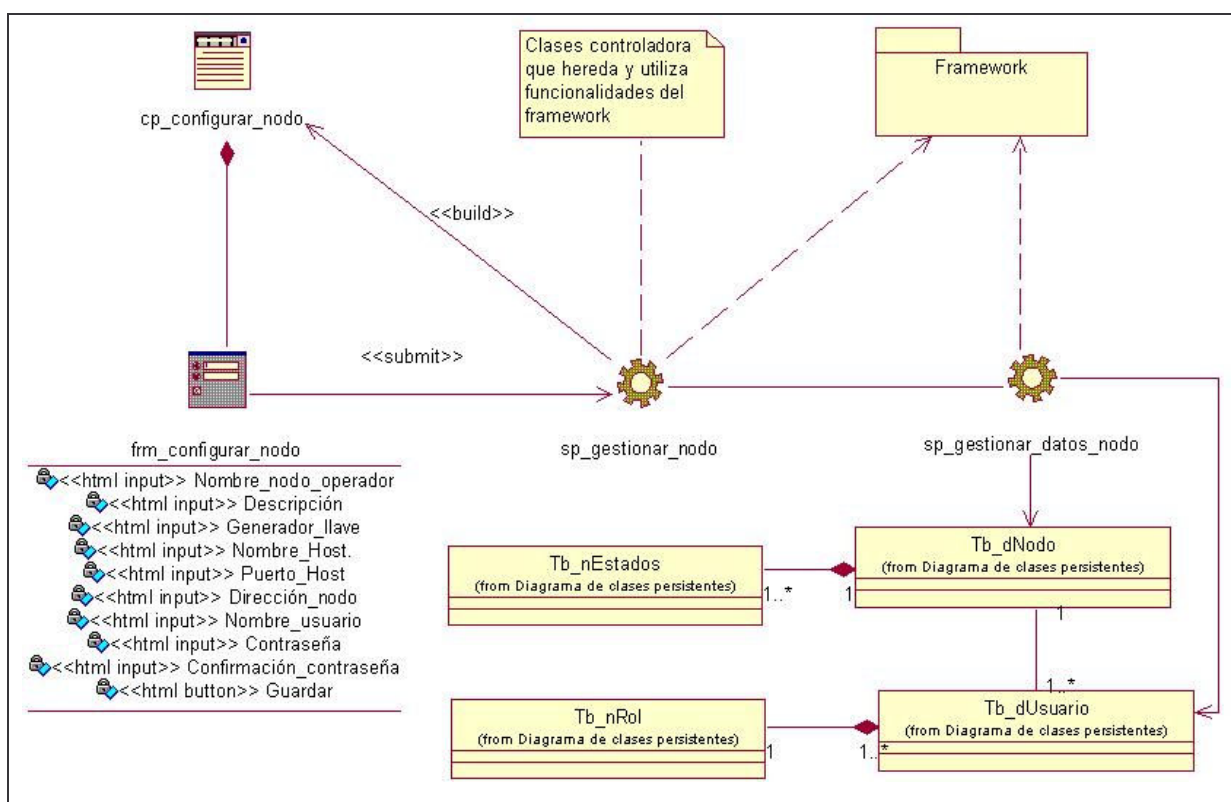


Figura 3.13 Diagrama de clases de diseño del CU Configurar Nodo.

TABLAS

Tabla 1 Descripción de las tablas de la Base de datos.

Nombre: Tb_bEntidad_Empresa		
Descripción: Datos generales de la empresa proveedora de los servicios.		
Atributo	Tipo	Descripción
id_empresa	varchar(41),not null	Identificador único para una empresa, dentro del registro UDDI.
nombre_emp	varchar(255) , not null	Nombre de la Empresa.
nombre_usuario	varchar(255),not null	Nombre del personal autorizado dentro de la empresa, a realizar

		cambios.
lenguaje	varchar(5),null	Define el lenguaje en que estarán los datos de la empresa.
descripcion	:varchar(255), not null	Descripción de la línea de trabajo que sigue la empresa.
nombre_nodo	varchar(255),not null	Nombre del Nodo Privado UDDI, desde donde se insertó la empresa.
ultima_actualizacion	timestamp,not null	Instante de tiempo en que la entidad tuvo un cambio.
firma_digital	varchar(256), not null	Este atributo permite encriptar la información a través de firmas digitales producidas por XML.
es_habilitado	bool	Indica si la empresa ha sido habilitada o deshabilitado.
es_revisado	bool	Indica si los datos de la empresa han sido revisados o no.
id_nodo	varchar(41), null	Identificador único para el nodo del registro UDDI.

Tabla 1.1 Entidad Empresa.

Nombre: Tb_dEmpresa_Identificador		
Descripción: Describe las distintas identificadores que puede tener la empresa.		
Atributo	Tipo	Descripción
id_identificador_emp	varchar(41),not null	Identificador de la entidad.
nombre_identificador_emp	varchar(255,null)	Nombre del el identificador.
valor_identificador_emp	varchar(255), not null	Valor del identificador.

Tabla1.2 Entidad Identificador de Empresa

Nombre: Tb_dEmp_Categoria		
Descripción: Describe las distintas categorías que puede tener la empresa. Ejemplo: categoría de acuerdo a la situación geográfica, categoría de productos, categoría de industria.		
Atributo	Tipo	Descripción
id_categoria_emp	varchar(41),null	Identificador de la entidad.
nombre_categoria_emp	varchar (255), null	Nombre de la categoría de empresa.
descripcion_categ_emp	varchar (255), not null	Valor de la empresa.

Tabla 1.3 Entidad Categoría de Empresa

Nombre: Tb_URL_Empresa		
Descripción: Es una lista de Localizadores de Recursos Uniformes (URL), que apuntan a la dirección en la red que tiene la empresa.		
Atributo	Tipo	Descripción
id_url	int, not null	Identificador para la dirección de la empresa
tipo_uso	varchar (255), not null	Es el tipo de URL, que tiene la dirección: TBusiness, homepage.
url	varchar (255), not null	Valor de la URL
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del registro UDDI.

Tabla 1.4 Entidad Empresa.

Nombre: Tb_dContactos		
Descripción: Contactos de el representante de la empresa.		
Atributo	Tipo	Descripción
id_contacto	int, not null	Identificador del contacto del representante de la empresa.

tipo_uso	varchar(255), not null	Define el tipo de contacto: cuestiones técnicas, contacto técnico, contacto de ventas, soporte técnico.
nombre_persona_contacto	varchar(255), not null	Especifica el nombre de la persona correspondiente al contacto.
apellido1	varchar(255), not null	Especifica el primer apellido de la persona correspondiente al contacto.
apellido2	varchar(255), not null	Especifica el segundo apellido de la persona correspondiente al contacto.
lenguaje	varchar (5), null	Define el lenguaje con que será escrito el contacto.
descripcion	varchar(255), not null	Breve descripción del contacto.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.

Tabla 1.5 Entidad Contactos de Empresa

Nombre: Tb_dDireccion		
Descripción: Representa la dirección física de la empresa.		
Atributo	Tipo	Descripción
id_direccion :	int, not null	Identificador para la dirección de la empresa.
tipo_uso	varchar(255), null	Describe el tipo de la dirección de contacto. Ejemplo: Oficinas centrales, la facturación la sección.
direccion	varchar(100), null	Contacto de la persona.
id_contacto	int, not null	Identificador del contacto del representante de la empresa.
id_empresa	varchar(41), null	Identificador único para una empresa,

		dentro del nodo del registro UDDI.
--	--	------------------------------------

Tabla 1.6 Entidad Dirección Física de la Empresa

Nombre: Tb_Direccion_Opcional.		
Descripción: Describe una dirección opcional en caso de que la dirección tenga otro formato diferente al normal, que se especifica en la entidad, Tb_dDirección.		
Atributo	Tipo	Descripción
id_dir_opc	int, not null	Identificador para la entidad de la dirección opcional.
direccion	varchar(100), null	Dirección opcional.
id_direccion	int, not null	Identificador para la dirección de la empresa.
id_contacto	int, not null	Identificador del contacto del representante de la empresa.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.

Tabla 1.7 Entidad Línea de Dirección de la Empresa

Nombre: Tb_Telefono		
Descripción: Define los teléfonos con los que se puede contactar al representante de la empresa.		
Atributo	Tipo	Descripción
id_telefono	int, not null	Identificador de la entidad teléfono.
tipo_uso	varchar(255), null	Describe los propósitos con los que se puede contactar ese teléfono.
num_telefono	varchar(50), not null	Número de teléfono.
id_contacto	int, not null	Identificador del contacto del

		representante de la empresa.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.

Tabla 1.8 Entidad Teléfono

Nombre: Tb_dCorreo_Eletronico		
Descripción: Define los correos electrónico con los que se puede contactar la representante de la empresa.		
Atributo	Tipo	Descripción
id_correo	int, not null	Identificador del correo electrónico.
tipo_uso	varchar(255), null	Describe los propósitos con los que se puede contactar ese correo electrónico.
direc_correo	varchar(255), not null	Dirección correo electrónico.
id_contacto	int, not null	Identificador del contacto del representante de la empresa.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.

Tabla 1.9 Entidad Correo Electrónico.

Nombre: Tb_dEntidad_Servicio		
Descripción: Describe las características que tiene el servicio Web registrado.		
Atributo	Tipo	Descripción
id_servicio	varchar(41), not null	Identificador único para un servicio, dentro de una empresa.
nombre	varchar(255), not null	Nombre del servicio.
lenguaje	varchar(5), null	Define el lenguaje en que estarán los datos del servicio.

descripcion	varchar(255), not null	Breve descripción de lo que hace el servicio Web.
ultima_actualizacion	timestamp, not null	Instante de tiempo en que la entidad tuvo un cambio.
firma_digital	varchar(256), not null	Este atributo permite encriptar la información a través de firmas digitales producidas por XML.
es_habilitado	bool	Indica si el servicio está habilitado o no.
es_revisado	bool	Indica si los datos del servicio han sido revisados o no.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.
id_usuario	varchar(20), null	Identificador del proveedor del servicio.
id_usuario_gral	varchar(20), null	Identificador único del usuario general.

Tabla 1.10 Entidad Servicio

Nombre: Tb_dEntidad_Referencia		
Descripción: Las Referencias poseen detalles técnicos de cómo invocar el servicio Web y como este se construye.		
Atributo	Tipo	Descripción
id_referencia	varchar(41), not null	Identificador único para una referencia, dentro de un servicio.
lenguaje	varchar(5), null	Lenguaje en que estará la referencia del servicio web.
descripcion	varchar(255), not null	Breve descripción de la función

		de esta entidad.
url_pto_acceso	varchar(20), null	Contiene la dirección de red para invocar el servicio.
redireccion_hosting	varchar(255), null	Este atributo tiene valor cuando un servicio es remotamente ofrecido (a través de un servidor, obteniendo el la dirección IP, nombre del servidor o por el protocolo XML: RPC) o cuando muchas descripciones del servicio podían beneficiarse de una descripción del servicio sola. Si tiene valor este atributo, el punto de acceso es nulo.
ultima_actualizacion	timestamp, not null	Instante de tiempo en que la entidad tuvo un cambio.
firma_digital	varchar(256), not null	Este atributo permite que se encripte la información a través de la firmas digitales producidas por XML.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.
id_servicio	varchar(41), not null	Identificador único para un servicio, dentro de una empresa.

Tabla 1.11 Entidad Referencia

Nombre: Tb_nTipo_Pto_Acceso		
Descripción:		
Atributo	Tipo	Descripción
id_tipo_pto_acceso :	integer(15), not null	Identificador del tipo de punto de

		acceso.
nombre	varchar(20), not null	Nombre del tipo de punto de acceso. (HTTP, SMTP, FTP). Si tiene valor este atributo, el redirección_hosting es nulo.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.
id_servicio	varchar(41), not null	Identificador único para un servicio, dentro de una empresa.
id_referencia	varchar(41), not null	Identificador único para una referencia, dentro de un servicio.

Tabla 1.12 Entidad Punto de Acceso

Nombre: Tb_dRef_Categoria		
Descripción: Estas categorías se refieren a características específicas de las Referencias, y describen el estado en que puede estar el servicio descrito por esta entidad; por ejemplo el estado de la certificación (prueba, producción) o versiones.		
Atributo	Tipo	Descripción
id_categoria_ref	varchar(41), null	Identificador para la entidad Categoría de la Referencia.
nombre_categoria_ref	varchar (255), null	Nombre de la categoría de Referencia.
valor_categoria_ref	varchar (255), not null	Valor de la categoría de Referencia.

Tabla 1.13 Entidad Categoría de la Referencia del Servicio.

Nombre: Tb_dInformac_Instancia_Especif_Tecnica
Descripción: Representa detalles de la especificación de la entidad Tb_dEntidad_Referencia, por cada referencia de una Especificación Tecnica(Tb_dEntidad_Especificacion_Tecnica).

Atributo	Tipo	Descripción
id_informac_instancia_especific_tecnica	int, not null	Identificador único para la información de la instancia de la especificación técnica del servicio.
lenguaje	varchar(5), null	Lenguaje en que esta la información de instancia de la especificación técnica del servicio.
descripcion	varchar(255), not null	Breve descripción de la funcionalidad de esta entidad.
vision_gral_url	varchar(255), null	Dirección donde esta el documento general que contiene los detalles de la especificación técnica del servicio, descrito a través del lenguaje descriptivo WSDL que utiliza XML.
tipo_vision_gral_url	varchar(20), not null	Provee la información sobre el tipo del documento que pertenece a la dirección del atributo vision_gral_url: texto, WSDL Interface
parametros_instancia	varchar(255), null	Es un documento XML que especifica los parámetros de instancia de la especificación técnica.
id_especificac_tecnica :	varchar(41), not null	Identificador único para una identificación técnica dentro de una entidad Tb_dEntidad_Referencia
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.
id_servicio	varchar(41), not null	Identificador único para un servicio, dentro de una empresa.

id_referencia	varchar(41), not null	Identificador único para una referencia, dentro de un servicio.
---------------	-----------------------	---

Tabla 1.14 Entidad Información de la instancia de la especificación técnica del servicio.

Nombre: Tb_dEntidad_Especificacion_Tecnica		
Descripción: Le especificación técnica es un recurso para determinar la compatibilidad con los servicios Web. A través de esta entidad, se describen los servicios Web. Gracias a las especificaciones técnicas las personas y los programas pueden descubrir cómo interactuar con servicios Web sobre los que no saben mucho.		
Atributo	Tipo	Descripción
id_especificac_tecnica :	varchar(41), not null	Identificador único para una identificación técnica dentro de una entidad Tb_dEntidad_Referencia
nombre_nodo	varchar(255), not null	Nombre del Nodo Privado UDDI, desde donde se insertó la Especificación Técnica.
nombre_esp_tecn	varchar(255), not null	Nombre de la Especificación Técnica.
lenguaje	varchar(5), null	Lenguaje con que está escrita la Especificación Técnica.
descripcion	varchar(255), not null	Breve descripción de lo que realiza la Especificación Técnica.
vision_gral_url	varchar(255), null	Dirección donde esta el documento general que contiene los detalles de la especificación técnica del servicio, descrito a través del lenguaje descriptivo WSDL que utiliza XML.
es_eliminado	bool	Indica si el elemento ha sido lógicamente eliminado
ultima_actualizacion	timestamp, not null	Instante de tiempo en que la

	null	entidad tuvo un cambio.
firma_digital	varchar(256), not null	Este atributo permite que se encripte la información a través de la firmas digitales producidas por XML.
es_habilitado	bool	Indica si la especificación técnica está habilitada o no
es_revisado	bool	Indica si los datos de la especificación técnica han sido revisados o no.
id_usuario	varchar(20), null	Identificador único del usuario.
id_usuario_gral	varchar(20), null	Identificador único del usuario general

Tabla 1.15 Entidad Especificación Técnica.

Nombre: Tb_dEspecificacion_Tecnica_Indentificador		
Descripción: Identificadores de las Especificaciones Técnicas.		
Atributo	Tipo	Descripción
id_ident_especificac_tecnica	varchar(255), null	Identificador de la entidad Identificador de Especificación Técnica.
nombre_ident_especif_tecnica	varchar(255), null	Nombre del identificador de la Especificación Técnica.
valor_ident_especif_tecnica	varchar(255), not null	Valor del identificador de la Especificación Técnica.

Tabla 1.16 Entidad Identificador de la especificación técnica.

Nombre: T_dEspecif_Tecnica_Categoria		
Descripción: Categorías de las Especificaciones Técnicas.		

Atributo	Tipo	Descripción
id_categ_especificac_tecnica	varchar(255), null	Identificador de la entidad Categoría de Especificación Técnica.
nombre_categ_especif_tecnica	varchar(255), null	Nombre de la categoría de Especificaciones Técnicas.
valor_categ_especif_tecnica	varchar(255), not null	Valor de la categoría de Especificaciones Técnicas.

Tabla 1.17 Entidad Categoría de la especificación técnica.

Nombre: Tb_dUsuario		
Descripción: Contiene los datos del proveedor.		
Atributo	Tipo	Descripción
id_usuario	varchar(20), null	Identificador único del usuario.
usuario_sistema	varchar(15), not null	Nombre único del usuario del sistema.
contraseña	varchar(15), not null	Contraseña que utiliza el usuario en el sistema para tener acceso a él.
tiene_cuenta_activa :	bool	Define si su cuenta es activa o cancelada.
creado	timestamp, not null	Tiempo en que se crea la cuenta del usuario.
id_empresa	varchar(41), null	Identificador único para una empresa, dentro del nodo del registro UDDI.
id_nodo	varchar(41), null	Identificador único para el nodo del registro UDDI.
id_rol	varchar(5), not null	Identificador del rol del usuario.

id_usuario_gral	varchar(20), null	Identificador único del usuario general.
-----------------	-------------------	--

Tabla 1.18 Entidad Usuario

Nombre: Tb_rHist_Usuario		
Descripción: Resultado de la relación muchos a muchos entre las entidades rol y usuario.		
Atributo	Tipo	Descripción
id_historial_usuario	varchar(51), not null	Identificador para el historial de cada usuario.
numero_usos	Int, not null	Número de veces que el usuario ha entrado al sistema.
id_usuario	varchar(20), null	Identificador único del usuario.
id_usuario_gral	varchar(20), null	Identificador único del usuario general.

Tabla 1.18 Relación Historial-Usuario

Nombre: Tb_dHistorial_Usuario		
Descripción: Controla las acciones del usuario dentro del sistema.		
Atributo	Tipo	Descripción
id_historial_usuario	varchar(51), not null	Identificador para el historial de cada usuario.
tiempo_sesion	timestamp, not null	Hora en que entra o sale del sistema.
dia_sesion	date, not null	Día en que entra o sale del sistema.
estado_sesion	bool	Acción que realiza, puede ser entrar o salir.

Tabla 1.19 Entidad Autenticación.

Nombre: Tb_nRol

Descripción: Es una tabla de estados que muestra el rol del usuario.		
Atributo	Tipo	Descripción
id_rol	varchar(5), not null	Identificador del rol del usuario.
nombre	varchar(5), not null	Nombre del rol.

Tabla 1.20 Entidad Rol de usuario.

Nombre: Tb_dSolicit_Usuario		
Descripción: Contiene los datos que realiza el usuario que solicita entrar datos de una empresa por primera vez al registro UDDI.		
Atributo	Tipo	Descripción
id_usuario	varchar(20), null	Identificador único del usuario.
nombre_empresa	varchar(255), not null	Nombre de la Empresa.
cargo_empresa	varchar(50), null	Cargo del solicitante dentro de la empresa.
url_empresa	varchar (255), not null	Dirección de la empresa en la red.
descripcion	varchar(300), null	Breve descripción de la línea de trabajo que sigue la empresa.
id_usuario_gral	varchar(20), null	Identificador único del usuario general.

Tabla 1.21 Entidad Solicitud de usuario.

Nombre: Tb_dUsuario_Gral		
Descripción: Es la tabla que contienen la información común entre los usuarios del sistema y los usuarios que solicitan ser del sistema.		
Atributo	Tipo	Descripción

id_usuario_gral	varchar(20), null	Identificador único del usuario general.
nombre_usuario	varchar(255), not null	Nombre del usuario.
apellido1	varchar (100), not null	Primer apellido del usuario.
apellido2	varchar (100), not null	Segundo apellido del usuario.
direc_correo	varchar(5), null	Correo electrónico del usuario.

Tabla 1.22 Entidad Usuario General.

Nombre: Tb_dNodo		
Descripción: Es la tabla que contienen la información del nodo del registro UDDI.		
Atributo	Tipo	Descripción
id_nodo	varchar(41), null	Identificador único para el nodo del registro UDDI.
nombre_nodo	varchar(255), not null	Nombre del Nodo Privado UDDI
descripcion	varchar (255), not null	Breve descripción de la funcionalidad del nodo.
nombre_host	varchar(5), not null	Nombre del host usado por el nodo para modelar los servicios.
puerto_host	Int, not null	Número del host usado para tener acceso a servicios de nodo.
url_nodo	varchar(255), not null	Dirección del nodo.
generador_llave	varchar(255), not null	Prefijo de las llaves de las entidades del nodo.
nombre_version	varchar(255), not null	Nombre de la versión de las Especificaciones UDDI que utiliza el

		Nodo Privado.
--	--	---------------

Tabla 1.22 Entidad Nodo.

Nombre: Tb_nEstados		
Descripción: Es la tabla que contienen la información del nodo del registro UDDI.		
Atributo	Tipo	Descripción
id_estado	varchar(41), null	Identificador único para del estado del nodo.
nombre_estado	varchar(255), not null	Nombre del estado del nodo.

Tabla

1.23

Entidad

Estado.

GLOSARIO DE TÉRMINOS Y SIGLAS.

Administrador: Es la persona que tiene privilegios para determinadas funcionalidades del sistema.

API: Interfaz de Programación de Aplicaciones. Es el conjunto de funciones y métodos que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción. Una API representa un interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas librerías que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación.

Arquitectura Cliente/Servidor: Modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

Categoría: Un valor que representa una división definida de la clasificación de un sistema especificado de la categoría.

Editor: Es la persona encargada de revisar, habilitar y deshabilitar las empresas, servicios, especificaciones técnicas e instancias de ellas; que los proveedores inserten en UDDI.

Entidad: Estructuras de datos.

Fax: Es un sistema que permite transmitir a distancia por la línea telefónica escritos o gráficos (telecopia).

Firma Digital: UDDI utiliza la firma digital para encriptar la información a través de las firmas digitales producidas por XML. Las firmas digitales se usan para suministrar integridad y autenticidad de los datos dirigidos en un Registro de UDDI.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

FTP: *File Transfer Protocol*. Es un protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos

podemos conectar a un servidor para descargar ficheros desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

GUID: Globally Unique Identifier (Identificador Global Único). Es un único número de 128-BIT generado por Windows o por algunas aplicaciones de Windows para identificar un componente, aplicación, fichero, entrada de una base de datos o un usuario.

HTML: *Hypertext Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTTP: *Hypertext Transfer Protocol*. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

IBM: International Business Machines o IBM, conocida coloquialmente como el Gigante Azul, es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

Identificador: Valor que representa una división definida de la identificación de un sistema especificado de identificadores.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

Intranet (Intranet, Intranet): Red propia de una organización, diseñada y desarrollada siguiendo los protocolos propios de Internet, en particular el protocolo *TCP/IP*. Puede tratarse de una red aislada, es decir no conectada a Internet

Linux: Es el nombre de un núcleo, pero se suele denominar con este nombre a un sistema operativo de libre distribución software libre (y de código abierto), donde el código fuente está disponible

públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.

Macromedia Dreamweaver MX: Herramienta para el desarrollo de aplicaciones Web de Macromedia. Combina en un único entorno de desarrollo accesible y potente las reconocidas herramientas de presentación visual de Dreamweaver, las características de rápido desarrollo de aplicaciones Web de Dreamweaver UltraDev y ColdFusion Studio, y el extenso soporte de edición de código de HomeSite. Ofrece una completa solución abierta para las tecnologías Web y estándares de hoy, incluyendo la accesibilidad y servicios Web.

Meta Datos: Datos que sirven para describir algo, por ejemplo, puede ser un registro que almacene información de un dato dado.

Microsoft: (acrónimo Microcomputer Software). Es una empresa de Estados Unidos, dueña y productora de los sistemas operativos: Microsoft DOS y Microsoft Windows, que se utilizan en la mayoría de las computadoras del planeta.

MVC: *Modelo Vista Controlador.*

NAICS: Sistema norteamericano de clasificación de la industria.

Nodo: Una colección de servicios Web, que implementa el conjunto de API de UDDI, las cuales permiten buscar y publicar información y se utilizan acorde al conjunto de políticas. Manejan un sistema particular de datos.

Páginas Amarillas en Internet: Es un portal comercial que tiene como objetivo, proporcionar información que anuncian los productos y/o servicios que se buscan. También brinda información útil que facilita las actividades personales y profesionales.

Páginas Blancas en Internet: Para encontrar a familiares, amigos o personas que sea de interés localizar. Incluye nombre, teléfono, y dirección.

Páginas Verdes en Internet: Páginas Verdes, directorio de empresas de consultoría, asesoría, medio ambiente

Parser: En informática es un analizador sintáctico. Es así mismo un programa que reconoce si una o varias cadenas de caracteres forman parte de un determinado lenguaje, es utilizado por ejemplo en compiladores.

PHP: *Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

Políticas: Definen el comportamiento y las reglas que debe seguir el registro UDDI y cada nodo en particular

PostgreSQL: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

Proveedor: Es la persona encargada de publica Empresas, Servicios y Especificaciones técnicas en UDDI.

Registro: Estructura compleja que define UDDI para descubrir e interactuar con los Servicios Web. Esta estructura está compuesta por cuatro tipos de datos básicos para la información del negocio y del servicio.

Registro UDDI: Es el conjunto de nodos.

RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

Sección Blanca en UDDI: Similar a la información que aparece en el directorio telefónico e Internet.

Sección Amarilla en UDDI: Similar a su equivalente telefónico, e incluyen categorías de catalogación industrial tradicionales, ubicación geográfica, etc. Mediante el uso de códigos y claves predeterminadas, los negocios se pueden registrar y así facilitar a otros servicios la búsqueda usando estos índices de clasificación.

Sección Verde en UDDI: Contiene la información técnica acerca de los servicios ofrecidos por los negocios. Se incluyen referencias de especificaciones de servicios Web así como también información complementaria para los mecanismos diversos de búsqueda basados en URL.

Servicio Web: Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Sistema de Categorías: Cada categoría en un sistema de categorías representa típicamente medios para agrupar entidades distintas con características similares.

Sistemas de Identificadores: Cada identificador en un sistema de identificadores representa típicamente medios para agrupar entidades distintas con características similares.

Sistemas Distribuidos: Es aquel en aquellos componentes, tanto de hardware como software, localizados en computadores, conectados en Red, comunican y coordinan sus acciones únicamente mediante el paso de mensajes.

Sistema Taxonómicos: Son sistemas creados para la clasificación y categorización de ciertos datos o informaciones.

SMTP: Simple Mail Transfer Protocol. Protocolo simple de transferencia de correo electrónico. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos

SOAP: *Simple Object Access Protocol*. Es un protocolo que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

Sockets: Concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada. Un *socket* queda definido por una dirección IP, un protocolo y un número de puerto. Un socket es un fichero existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

Software (componentes lógicos, programas, software): Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

Sitio Web: Es un conjunto de páginas Web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.

Token: Son identificadores que se pueden utilizar para consultar o manipular el inicio de sesión. Una vez que se obtiene un token, se puede averiguar la identidad del usuario que representa el inicio de sesión y determinar si es posible concederle acceso a los recursos protegidos.

UCI: *Universidad de las Ciencias Informáticas.*

UDDI: Universal Description, Discovery and Integration. Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones y a los usuarios comprobar qué servicios Web están disponibles.

UML: *Unified Modeling Language.* Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

URI: Un identificador Uniforme del Recurso.

URL: *Uniform Resource Locator* (Localizador Uniforme de Recursos) Sistema unificado de identificación de recursos en la red. Las direcciones se componen de protocolo, FQDN y dirección local del documento dentro del servidor. Este tipo de direcciones permite identificar objetos *WWW*, *Gopher*, *FTP*, *News*,... Ejemplos de URL son: <http://cubasi.cu> o <ftp://ftp.ati.es>

UNSPSC: Clasificación universal estándar de productos y servicios.

Usuario: Es la persona encargada de buscar informaciones de empresas, servicios y especificaciones técnicas que se publican en UDDI.

WEB (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

WML: Website Meta Language.

WSDL: *Web Services Description Language.* Es un formato XML que se utiliza para describir servicios Web.

WSIL: Web Service Inspection Language.

WS-Security: Seguridad en Servicios Web. Es un protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web.

XML: *Extensible Markup Language*. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.